N° 74 mensuel Novembre-Décembre 1991-24 F

La Revue de l'Utilisateur PC

Le magazine professionnel en programmation PC

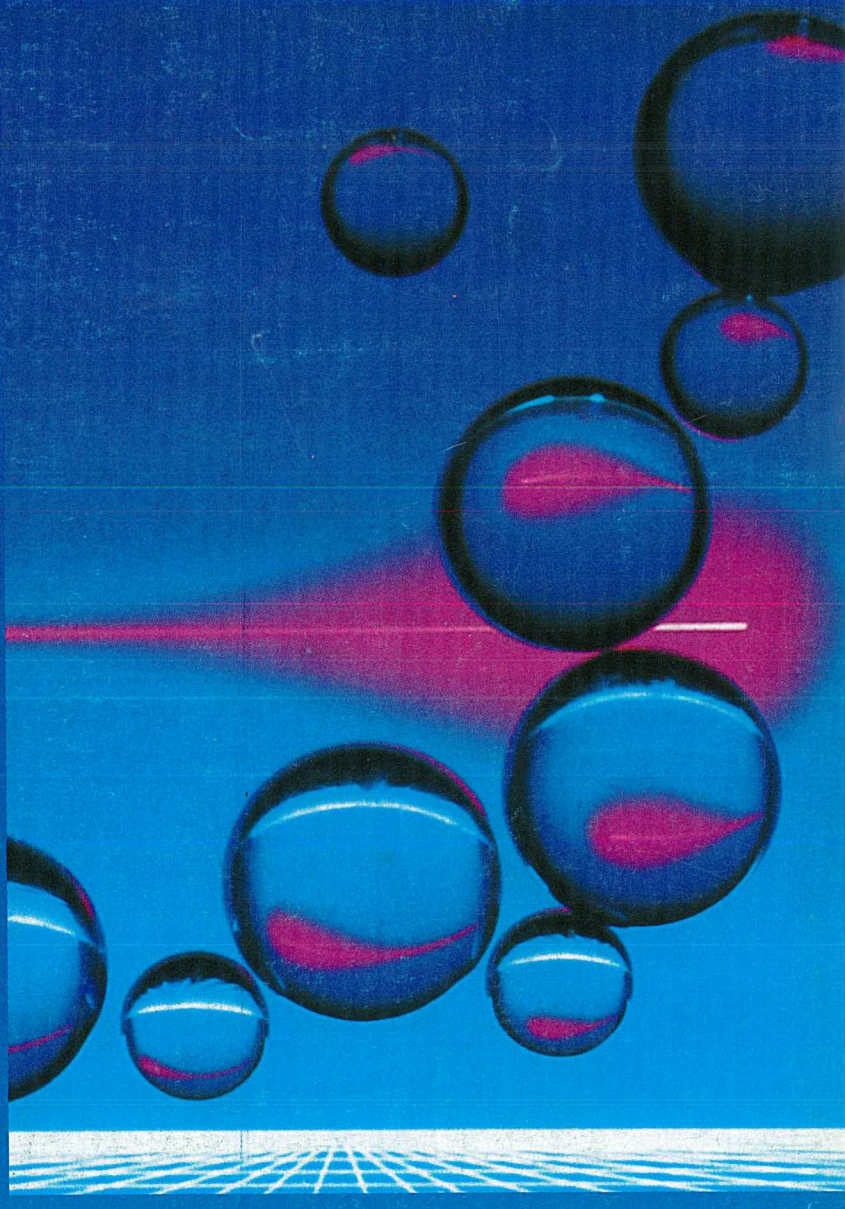
UNE LIBRAIRIE de compression de données pour C, Clipper, Pascal, Basic, ASM The Data Compression Library

WINDOWS et le paging

CLIPPER 5.01, TBROWSE : les couleurs

CALCULATRICE CALC.C

VGA s'imposant comme le ciment de toutes les applications graphiques, nous allons lui construire une librairie en béton



NOVEMBRE-DÉCEMBRE 1991 - N° 74 - 24 F - Suisse 8 FS - Belgique 200 FB - Canada 6,25 $ Can.

M 2537 - 74 - 24,00 F -RD

LOGICIELS DERNIERES VERSIONS
POUR PC, XT, AT, COMPATIBLES,
UNIX, XENIX ET MACINTOSH AUX

MEILLEURS PRIX

SOFTWORLD

17, Avenue Emile Zola, 75015 Paris. (Métro Javel)
Téléphone : (1) 40 59 02 99 Fax : (1) 45 79 95 55

Pour MACINTOSH

		Prix Public TTC	Notre prix TTC
BASES DE DONNEES			
4ème Dimension	f	~~7353~~	5885
File Force	f	~~3321~~	2250
FoxBase Mac Mono	æ	~~6820~~	5550
Omnis 5 *Blyth*	f	~~7110~~	6399
PAO / DESSIN			
Page Maker 4.0 *Aldus*	f	~~9607~~	7250
Canvas 3.0 *Deneba*	a	~~4140~~	3290

MAPLE V

Un outil intégral de calcul numérique et formel, de représentation graphique 2D ou 3D et de programmation symbolique interactive de haut niveau pour Scientifiques, Mathématiciens.
Version Mac ou autres systèmes
Version Mac : 4500 HT ou 5337 TTC

TABLEURS			
Excel 3.0 *Microsoft*	f	~~4732~~	3295
WINGZ Avec graphe 3D	f	~~5325~~	4350
EDITEURS DE TEXTE			
Word 4 *Microsoft*	f	~~3546~~	2450
Formulator *(Mathématiques)*	a	~~1950~~	1490
INTEGRES			
Works *Microsoft*	f	~~2953~~	2250
LANGAGES			
Think C 4.0	**a**	**~~2953~~**	**1595**
Think Pascal 3.0	**a**	**~~2953~~**	**1595**
Mac SchemeLisp+Tools.	a	nc	4250
Prolog 2.0 *Adv.A.I. Syst.*	a	~~3530~~	2950
Professional Programmer's Ext.	a	nc	2995
Prototyper (Générateur de Codes sources)	a	nc	1995
QuickBasic *Microsoft*	a	~~1175~~	895
Smalltalk V/Mac *Digitalk*	a	~~1767~~	1503
Turbo Pascal *Borland*	f	~~1180~~	850
Turbo Pascal DataBase	a	~~1180~~	850
Turbo Pascal Numeric	a	~~1180~~	850
Turbo Pascal Tutor *Borland*	a	~~1180~~	850
ZBasic 5.0 *Zedcor*	a	~~2022~~	1395
SCANNER ET O.C.R.			
Omnipage (O.C.R.)	a	~~11563~~	9750
Scanner couleur *Microtek*	a	~~29532~~	24900
UTILITAIRES			
Mac Tools Deluxe *Central P.*	a	~~1529~~	1095
SAM 3.0 (Antivirus)	a	~~1175~~	990
Virtual 030 pour Mac 68030	a	~~2360~~	1895

INGENIERIE

ACNAP Analyseur circuits AC	a	nc	4550
DCNAP Analyseur circuits DC	a	nc	2600
MICRO-CAP III (PC seulement)	a	nc	14950
SPP Traitement des signaux	a	nc	1950
Systat 5.0 Statistiques-Graphiques	a	nc	7950

MUSIQUE			
MacRecorder *Farallon*	a	nc	2150
Professional Composer *Mark Uni.*	a	~~6227~~	4950

a : Version en américain.
f : Version en français.
æ : en américain avec possibilité d'échange en français.
nc : non communiqué

TOUS NOS PRIX SONT TTC
PRIX INDICATIF AU 2 SEPTEMBRE 1991
Extrait de notre catalogue
Catalogue gratuit sur demande

POUR PC COMPATIBLES

		Public TTC	Notre TTC
ADA et Librairies			
Janus ADA Compiler +Tlkt	a	~~5871~~	4950
JanusADACompiler 386+Tkt	a	~~5871~~	4950
ASM et Librairies			
386 DOS Extender *Pharlap*	a	nc	4950
Btrieve 5.00C *Novell*	a	nc	5950
EMS Toolkit *Realtime H.*	a	nc	1350
Intel ASM-386 Macro ASM	a	nc	6000
Intel ASM-86 Soft.Dev.Kit	a	nc	7500
MASM 5.1 *Microsoft*	a	~~1767~~	1350
PCX Effect + Source *Genus*	a	nc	1895
PCX Prog's Tlkit+Code Srce	a	nc	5995
PCX Text + Source *Genus*	a	nc	2895
BASIC et Librairies			
Basic 7.1 *Microsoft*	a	~~4732~~	3795
Basic Develop. Tools *Sterli.*	a	nc	1175
Btrieve *5.0 Novell*	a	~~2906~~	2550
FABS PLUS *Computer C*	a	nc	2350
Graph Pak Pro *Crescent*	a	nc	1773
Logic Gem *Sterling Castle*	a	nc	1150
MicroHelp Toolbox	a	nc	1195
PDQ *Cresent*	a	nc	1295
Toolbox II *MicroHelp*	a	nc	1250
PCX Effects + Source *Genus*	a	nc	1895
PCX Prog's Tlkit+Code Srce	a	nc	5995
PCX Text + Source *Genus*	a	nc	2895
Realizer *Within Technology*	a	~~4685~~	3950
Sciences & Engin. Tools	a	nc	1195
Quick Basic 4.5 *Microsoft*	f	~~1175~~	895
QuickPak Windows *Crescent*	a	nc	1990
QuickPak Pro QB 4.5 *Cresce.*	a	nc	2005
QuickPak Scientific *Crescent*	a	nc	1502
QuickScreen QB4.5 *Crescent*	a	nc	1502
QB Vol. 6 *MicroHelp*	a	nc	1295
Scientific SubroutineLibrary	a	nc	2250
Stay-res + *MicroHelp*	a	nc	1295
Turbo Basic *Borland*	æ	~~1185~~	825
Turbo Basic Toolbox chaque	æ	~~1180~~	825
Visual Basic *Microsoft*	f	~~2360~~	1770
C, C++ et Librairies			
C Asynch Manager *Blaise*	a	nc	1890
C Compiler 6.0 *Microsoft*	a	~~4495~~	3250
C Tools Plus *Blaise Comput*	a	nc	1495
GFX Graphics Library	a	nc	2135
GFX Fonts & Menus Lib.	a	nc	2135
GFX Font Editor	a	nc	949
GFX Screen Dump	a	nc	949
GreenLeaf Data Windows	a	nc	4250
Halo *Pro Media Cybernetics*	a	nc	5396
High C 386 *MetaWare*	a	nc	8950
Lattice C *Lattice*	a	~~5812~~	4950
Power Screen *Blaise Comput*	a	nc	1495
Quick C 2.5 *Microsoft*	a	~~1175~~	895
Super Functions *Greenleaf*	a	nc	2950
Turbo C Tools *Baise Comp*	a	~~2123~~	1495
Turbo C++ 1.0 *Borland*	f	~~1773~~	1250
Turbo C++ 1.0 Pro *Borland*	f	~~3552~~	2350
Turbo C++ RuntimeLibrary	a	~~1773~~	1495
Win++ *Blaise Computing*	a	nc	2495
Zortech C++ 2.1 *Zortech*	a	~~1773~~	1450
Zortech C++ 2.1 Developper	a	~~4501~~	2995

		Public TTC	Notre TTC
COBOL			
Cobol 4.0 *Microsoft*	a	~~8290~~	5950
Cobol / 2 *Microfocus*	a	nc	18000
RM Cobol Dvt *Mac Farland*	a	nc	12500
DEBUGGEURS			
Intel ASM-386 Soft.Dev.	a	nc	7500
MultiScope - Windows	a	nc	3995
Periscope IV (25 MHz)	a	nc	25950
View232 *Blaise Computing*	a	nc	1890
EDITEURS			
Brief 3.0 *Solution Systems*	f	~~4495~~	2250
EC Editor *C Source*	a	nc	895
Epsilon *Lugaru Software*	a	nc	1950
Norton Editor *Peter Norton*	æ	nc	750
VEdit Plus *CompuView*	a	nc	1850
GENERATEURS			
Automated Programmer*KGK*	a	nc	4950
Matrix Layout pr Basic, C, Pascal	a	nc	1950
PCYACC Corporate*Abraxas*	a	nc	5500
INTELLIGENCE ART.			
PC SCHEME *Texas Instru.*	a	nc	1195
Smalltalk V *Digitalk*	a	nc	895
Smalltalk V/286 *Digitalk*	a	nc	1450
Smalltalk V/Windows *Digit.*	a	~~4732~~	4250
Communication Pak *Digitalk*	a	nc	425
EGA/VGA Pak *Digitalk*	a	nc	425
Goodies 1 ou 2 ou 3 *Digitalk*	a	nc	425
µuLisp 90 *Soft Warehouse*	a	nc	3558
PASCAL et LIBRAIRIES			
MTASK pro *Ramsi*	**f**	**~~2360~~**	**1950**
Pascal 4.0 *Microsoft*	a	~~4732~~	3695
Asynch Manager *Blaise C.*	a	nc	1750
Power Screen *Blaise C.*	a	nc	1490
Pascal View 232 *Blaise C.*	a	nc	1890
Quick Pascal *Microsoft*	**f**	**~~1175~~**	**895**
Turbo Pascal 6.0 *Borland*	**f**	**~~1773~~**	**1250**
Turbo Pascal Pro 6.0	**f**	**~~3552~~**	**2350**
AUTRES LANGAGES			
Actor *White Water Group*	a	~~10615~~	7950
Fortran 5.0 *Microsoft*	a	~~5990~~	3250
Guideline C++ *Guidelines*	a	nc	3950
DIVERS			
Dr DOS 6.0	f	~~919~~	735
Merge 386 2 Utilisateurs *Locus*	a	~~4151~~	3995
Nantucket Clipper 5.0	f	~~11800~~	9250
Norton Guides *Norton* Chque	a	~~1186~~	995
PC MOS/386 Mono *Soft.Link*	a	~~2953~~	2495
Window 3.0 *Microsoft*	f	~~2360~~	1650
VM 386 *IGS*	f	~~3499~~	3050
VM 386 Net Pak *IGS*	f	~~2965~~	2650
Windows Devel. Toolkit 3	a	~~4495~~	3295

UNIX
Systèmes / Langages / Fichiers / SGBD
Editeurs de Texte / Utilitaires
Nous contacter : (1) 40 59 02 99

		Public TTC	Notre TTC
BASES DE DONNEES			
dB XL *Wordtech systems*	f	~~4715~~	3850
Fox Base+ 2.10 *Fox Soft.*	f	~~9429~~	7500
Fox Pro Mono *Fox Soft.*	f	~~10615~~	8150
Genifer+Template CompilerFoxBas	a	nc	4195
Paradox 3.5 *Borland*	f	~~9963~~	7150
Super DB *Computer Asso.*	f	~~7021~~	3500
Super Base 4/Windows	f	~~7412~~	5411
CAO / DAO / PAO			
Design CAD 2D (3D +1000 Frs)	f	~~4685~~	3950
Page Maker 4.0 PC *Aldus*	f	~~9610~~	6750
Ventura 2.0 *Rank Xerox*	f	~~9192~~	8550
COMPTA/GESTION			
Ciel! Compta-Gestion *Ciel!*	f	~~1156~~	925
Compta Saari Major *Saari*	f	~~16905~~	13950
INTEGRES			
Ability Plus Dual *Migent*	f	~~1927~~	1695
Works Windows *Microsoft*	f	~~2950~~	1995
TABLEURS			
Compile 12C Compiler WK.	a	nc	8950
Excel PC (286 et 386)	f	~~5918~~	3895
Lotus 123 Version 3.0	a	~~6750~~	4950
Multiplan 4 *Microsoft*	f	~~3309~~	2395
Quattro Pro 2.0 *Borland*	f	~~5924~~	4150
VP Planer + 3D *Paperback*	f	~~3440~~	2595
TRAITEMENT/ TEXTE			
Chiwriter (Scientifiques)	f	~~1235~~	1195
MathEdit (Mathématiques)	a	~~2366~~	**2366**
Sprint 1.5 *Borland*	f	~~2959~~	2050
Word 5.5 *Microsoft*		~~5325~~	3550
Word pour Window *Micros.*	f	~~5918~~	4150
Wordperfect 5.1 *Wordperf.*	f	~~5693~~	4595
UTILITAIRES / SYST.			
Above Disc EMS 4.0	æ	~~1510~~	1195
CopyWrite *Quaid*	a	~~1175~~	695
Disk Explorer *Quaid*	a	~~1175~~	695
Norton Adv. Utilities 5.0	a	~~1779~~	1450
Norton AntiVirus	f	~~1175~~	950
PC Tools Deluxe 7.0	**f**	**~~2005~~**	**1395**
SOURCER+BIOS	**a**	**~~2075~~**	**1550**
MATHEMATIQUES			
Derive 2.07 *Soft Warehouse*	**a**	**~~2965~~**	**1928**
Meijin++ pour C++	a	~~3309~~	2790
Grapher *(Graphiques scientif.)*	a	~~2953~~	2366
HiWire Plus Wintek	a	nc	9500
Maple V DOS-386/486	a	nc	8243
MathCAD 2.5 *MathSoft*	a	~~6274~~	4950
Mathematica DOS-386/7 V2	a	nc	Tel
Mathematica DOS-386/Weitek V2	a	nc	Tel
QuickPak Scientific *Crescent*	a	nc	1502
Sciences & Engineer. Tools	a	nc	1195
Surfer *(Surface, carte géo. 3D)*	a	~~6111~~	4950
SYSTAT 5.0 + SYGRAPH	a	nc	8995
TKSolver	a	nc	4685

PRO-CITE

Gestion bibliographique. Version US pour Compatibles ou Macintosh
~~4684,70 FF TTC~~ PROMO 3950 FF TTC

BON DE COMMANDE
SOFTWORLD

17, Av. Emile Zola, 75015 Paris. (Métro Javel)
Téléphone : (1) 40 59 02 99 / Fax : (1) 45 79 95 55

Mon ordinateur est du type : (Macintosh) (PC Compatible)
Format de disquette : 3 1/2 ○ ou 5 1/4 ○
Je veux commander les produits notés ci-contre.
Mode de paiement :
○ Par Chèque ci-joint.
○ Par Contre remboursement +40 Frs (montant d'achat <5000 Frs)
○ Par Carte Bleue N° _____
Date d'expiration : _____
Date :
Signature : 1991

Nom : _____ Prénom : _____
Adresse : _____
_____
_____
Code postal : _____ Localité : _____
Téléphone : _____

Désignation	Quantité	Prix TTC
Frais d'envoi :		
Total TTC :		

Frais d'envoi recommandé obligatoire 40 Frs par 1000 Frs

Aucun règlement ne sera encaissé avant l'expédition des marchandises

Nous fournissons la plupart des produits français et étrangers à des prix défiant toute concurrence et dans un délai record.
Plus de 3000 produits pour PC et MAC.
Applications verticales, Bases de données, D.A.O., Editeurs de Textes, Graphiques, Ingénierie, Intégrés, Langages, Mathématiques, Librairies, P.A.O, Statistiques, Tableurs, Utilitaires...

Heures d'ouverture
10h à 12h et de 14h à 19h
Fermé Samedi , Dimanche et Fériés

EDITO

Le XIe Congrès de l'E.A.O. et du Multimédia ou comment la micro-informatique prend un nouveau virage

Le XIe Congrès de l'E.A.O. et du Multimédia, qu'organise notre confrère du Journal de la Formation Continue et de l'E.A.O. aura lieu les 7, 8 et 9 Janvier 1992 au Palais des Congrès, Porte Maillot (Paris 17e).

Les professionnels de la micro-informatique, les SSII, les développeurs notamment ne réalisent pas encore qu'il s'agit du plus grand événement de l'année dans le domaine de la micro-informatique.

L'avènement du Multimédia est en train de modifier tout ce que l'on croyait savoir en matière de PC.

Les fabricants de PC commencent à considérer le Multimédia comme la dernière révolution, certains disent parfois une nouvelle génération de la galaxie des micros.

Pour accompagner cette révolution, les géants de l'électronique grand public conçoivent déjà de nouvelles générations de CD vidéo, et, les agents de télécommunications y voient une source inépuisable de nouveaux services à valeur ajoutée.

La combinaison entre textes, graphiques, sons et images animées et télécommunications offre, à cet égard, des possibilités infinies de manipuler des données, à la fois pour créer des applicatifs nouveaux et à enrichir les progiciels existants.

Dans quelques années un traitement de textes ou un tableur ne seront plus tels que nous les connaissons aujourd'hui.

D'où l'apparition d'un marché extrêmement prometteur.

Ainsi que nous l'avions déjà annoncé dans le n° 66, (Novembre 1990), de la Revue de l'Utilisateur de l'IBM PC, ce marché représentera, en 1994, près de 65 Milliards de Francs.

Dès l'année prochaine, l'on comptera 6 Millions de CD-ROM, chaque CD-ROM étant l'équivalent de 1 500 disquettes ou pouvant contenir 650 000 pages de textes, 44 heures de son, 72 minutes d'un film.

Conséquences de cette révolution, la plupart des constructeurs et des éditeurs de logiciels se sont lancés dans le Multimédia : Philips et son CD-I (Compact Disc Interactive), Intel avec le DVI (Digital Video Interactive), Commodore et son système CDTV (Commodore Dynamic Total Vision), Microsoft avec Windows 3 et ses extensions, sans oublier bien sûr Apple avec Quicktime, extension du Multimédia du MacInstosh.

Du côté des constructeurs, Victor, Zénith, Siemens-Nixdorf, sans oublier ATT, Olivetti... ont rallié la norme MPC (Multimédia PC), et, Next, avec sa plate-forme avait déjà anticipé l'émergence du Multimédia.

Le XIe Congrès de l'E.A.O. et du Multimédia comportera une exposition qui réunira ces constructeurs ainsi que des développeurs de Multimédia et des éditeurs.

Deux colloques se tiendront en parallèle, dans le cadre du XIe Congrès de l'E.A.O. et du Multimédia, où constructeurs, développeurs, éditeurs et utilisateurs échangeront et confronteront leurs expériences et leurs standards.

C'est que le XIe Congrès de l'E.A.O. et du Multimédia à vocation « très business » contribuera à la recherche de normes, de standards et d'applications nouvelles du Multimédia.

Le monde de la micro-informatique va être bouleversé à l'horizon 1992 par le Multimédia, et, les développeurs se doivent de prendre en compte cette mutation à venir en participant au XIe Congrès de l'E.A.O. et du Multimédia.

Tous les informaticiens doivent prendre conscience de cette « révolution » de la micro-informatique professionnelle.

Pour obtenir le programme des colloques et des invitations pour visiter l'exposition il convient de s'adresser au Journal de la Formation Continue et de l'E.A.O.
2 rue d'Amsterdam, 75009 Paris
Tél. : (1) 47.64.07.57 - (1) 42.67.93.80 - (1) 42.67.92.99

Paul LAMBASLE

SOMMAIRE

4

WINDOWS ET LE PAGING

En mode 386 étendu, WINDOWS utilise de l'espace disque pour augmenter les ressources mémoire disponibles

Par Daniel VASSEUR

49

VGA S'IMPOSANT COMME LE CIMENT DE TOUTES LES APPLICATIONS GRAPHIQUES, NOUS ALLONS LUI CONSTRUIRE UNE LIBRAIRIE EN BETON

Le dopage d'une machine ancienne n'est pas une mince affaire : nombreux sont les pièges à déjouer et les astuces à connaître

Par Daniel VASSEUR

14

UNE LIBRAIRIE DE COMPRESSION DE DONNEES POUR C, CLIPPER, PASCAL, BASIC ASM THE DATA COMPRESSION LIBRARY

Tout le monde (ou presque) connaît le fameux utilitaire de compression de fichiers PKZIP, et combien d'entre vous auraient souhaité pouvoir intégrer un tel compacteur dans leurs applications ?

Par G. LEMEUNIER (OCIS)

55

CALCULATRICE CALC.C

Ce programme permet d'effectuer de longs calculs le plus simplement du monde. En effet, une page entière de calculs ne l'effraie pas

Par Laurent LEFOUET (ESGI)

32

CLIPPER 5.01 TBROWSE : les couleurs

Une approche de TBrowse par son côté le plus attrayant : les possibilités d'utilisation de la couleur dans des fenêtrages attractifs

Par Dominique BRUYERE

La Revue de l'Utilisateur PC

Le magazine professionnel en programmation PC

Mensuel 24 F
2, rue d'Amsterdam
75009 Paris
Tél. : (1) 42.81.54.27.
Service abonnement :
Tél. : (1) 42.81.54.27.

Directrice de la Publication :
Sardie ERRERA

Editeur :
Roger CHRISTOPHE

Rédacteur en chef :
Daniel VASSEUR

Directeur de fabrication :
Michel MILLET

Directeur de publicité :
Liliane ERRERA
Tél. : (1) 42.67.93.80.

Edité par SOFTin Communications (France) Sarl
R.C. Paris B 329 834 592
N° Siret
329 834 592 000 19
Commission Paritaire (en cours)
Dépôt légal :
4e trimestre 1991

CCP Paris 8850 84 D

Impression :
SNIL
93600 Aulnay-sous-Bois

Composition :
SOFTin Communications

Arrêtez de développer ce qui existe déjà...

COMPILEZ... C'EST GAGNÉ !

FINI

La programmation sans fin
Les séances de debugging
Les essais interminables

AGENDA Multi-Utilisateur

Cette gestion des rendez-vous très visuelle, vous sera très vite indispensable. Elle gère l'utilisation en multi-poste, permet la programmation des rendez-vous, de rechercher des plages de temps libres. Possibilité de couper-coller des rendez-vous.

- Version exécutable bridée..... **150 F H.T.**
- Version exécutable..... **649 F H.T.**
- Source en TP 6.0.....**2490 F H.T.**

CODE A BARRE

Impression des codes à barres au standard EAN 13, 2/5 entrelacé et Alpha 39.

- Version exécutable..... **649 F H.T.**
- Source en TP 5.5.....**1990 F H.T.**

GESTION DES STOCKS FACTURATION - CAISSE

Gestion des fichiers produits, clients, fournisseurs. Permet d'éditer vos devis, factures, avoirs, bons de commande, bons de livraison. Gère les ventes au "comptoir", édition des tickets de caisse, du ticket récapitulatif. La version Stock 4 comporte en plus une gestion du multiposte, une aide contextuelle, une gestion des représentants, les taxes parafiscales, un générateur d'état, des éditions paramétrables, des sélections multi-critères pour toutes les éditions (listes, courriers, fiches récapitulatives, etc...), import-export de fichiers en ASCII.

- Version exécutable de Stock 4 bridée **150 F H.T.**
- Version exécutable de Stock 3..... **430 F H.T.**
- Version exécutable de Stock 4.....**1850 F H.T.**
- Source Stock 3 en TP 3.0.....**2990 F H.T.**
- Source Stock 4 en TP 5.5.....**5990 F H.T.**

GESTION FICHIER COMMERCIAL

Gestion d'un fichier de prospection. Grâce à son générateur d'état intégré, il vous offre la possibilité de mailing ciblé et personnalisé et d'édition d'étiquettes. Le programme gère le multiposte, comporte une aide contextuelle, etc...

- Version exécutable bridée..... **150 F H.T.**
- Version exécutable.....**1260 F H.T.**
- Source en TP 5.5.....**3990 F H.T.**

SIMASOFT le concepteur et l'éditeur de logiciels le plus souple du marché, vous permet désormais d'acquérir les sources de plusieurs de ses programmes (écrits en Turbo Pascal). Vous aurez alors la permission de revendre vos nouvelles versions compilées sans limitations quantitatives et sans payer de redevances. Pour ceux qui ne possèdent pas le Turbo Pascal nous pouvons aussi vous personnaliser ces logiciels.

Nos logiciels sont utilisés par milliers dans de nombreux domaines (PME-PMI, artisans, commerçants, professions libérales, médicales, para-médicales), d'où une garantie de fonctionnement irréprochable.

Si vous ne connaissez pas nos logiciels, des versions bridées (déductible d'un achat ultérieur) vous permettront de juger de leurs qualités.

Nos logiciels fonctionnent sous MS-DOS (Version > = 3.0) sur un ordinateur compatible PC avec 640 K de RAM et un disque dur. Pour compiler les sources, vous devez posséder le compilateur Turbo Pascal de Borland.

Nos prix sont HT (TVA 18.6%). Une facture justificative est jointe aux envois. Les commandes doivent impérativement être accompagnées du règlement du montant total (logiciel+frais de port) soit par chèque, soit par mandat. Les frais de port sont de 29 F par commande, ou de 50 F par envoi en contre-remboursement, ou de 100 F pour les DOM-TOM et l'exportation.

Rédigez votre bon de commande de manière lisible, en indiquant vos nom, prénoms, adresse complète, Numéro de téléphone, ainsi que le format des disquettes que vous utilisez. Datez, signez et envoyez votre commande à l'adresse ci-dessous :

SIMA SOFTWARE

12, RUE DE LA RÉPUBLIQUE
13420 GÉMENOS - TÉL. 42.32.09.32
FAX. : 42.72.66.54 ET 42.32.17.27

COMPTABILITE GENERALE

Comptabilité générale multi-sociétés. Saisie des écritures par brouillard. Lettrage automatique et manuel. Editions des brouillards, journaux, grands-livres, balances, bilan, compte de résultat. La version II gère en plus le multiposte, des guides de saisies automatiques.

- Version exécutable bridée PROFIMAX II **150 F H.T.**
- Version exécutable PROFIMAX I.... **449 F H.T.**
- Version exécutable PROFIMAX II...**1690 F H.T.**
- Source PROFIMAX I en TP 3.0.....**3490 F H.T.**
- Source PROFIMAX II en TP 5.5....**5990 F H.T.**

COMPTABILITE LIBERALE 2035

Comptabilité simplifiée pour les professions libérales (soumises ou non à la TVA). Saisie en mode recettes - dépenses.

- Version exécutable bridée..... **150 F H.T.**
- Version exécutable.....**1690 F H.T.**
- Source en TP 6.0.....**5990 F H.T.**

GESTION DE LA PAYE ET DU PERSONNEL

Paye multi-sociétés entièrement paramétrable. Calcul de la paye mensuelle, horaire, paye à l'envers. Editions des bulletins aux nouvelles normes. Gestion des congés payés, des repos compensateurs, calcul de la CSG. Editions du livre de paye, des fiches individuelles, des fiches récapitulatives, de l'état préparatoire à la DADS, des avis de virements. Gestion des règlements, des acomptes.

- Version exécutable bridée..... **150 F H.T.**
- Version exécutable.....**1790 F H.T.**
- Source en TP 5.5.....**5990 F H.T.**

SENATOR II
GESTION COMPLETE D'ENTREPRISE

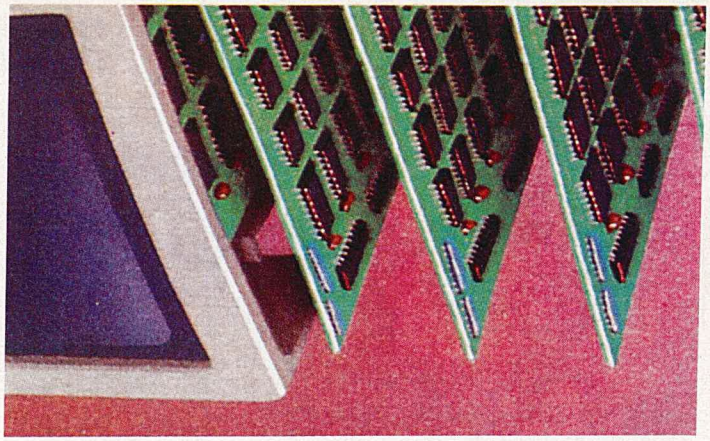
Reprend toutes les fonctionnalités du programme Stock 3 (GESTION DES STOCKS - FACTURATION - CAISSE), du programme Profimax I (COMPTABILITE GENERALE). De plus il contient un traitement de texte et une gestion de fichier commercial. Le tout est lié dans un ensemble cohérent et interactif (Transfert des factures, avoirs en comptabilité, transfert des prospect en gestion de stock, accès au traitement de texte pour toutes les éditions, Etc...

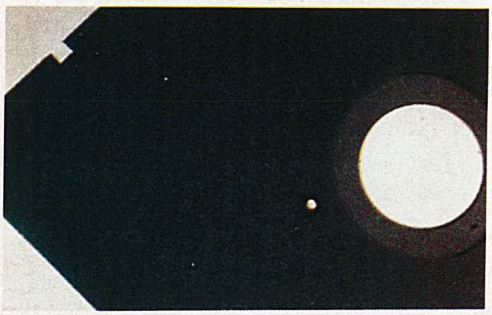
- Version exécutable.....**1260 F H.T.**
- Source en TP 3.0.....**6990 F H.T.**

VOUS ALLEZ GAGNER DE L'ARGENT

PAS DE REDEVANCES

WINDOWS et le paging





En mode 386 étendu, WINDOWS utilise de l'espace disque pour augmenter les ressources mémoire disponibles.

Prenons pour cobaye une machine de type 80386 équipée de 4 MO de RAM et consultons l'écran ''A propos de...'' dans le menu ''?'' du gestionnaire de programmes.

En fonction de différents critères que nous allons étudier tout au long de cet article, nous constatons que WINDOWS nous fait apparaître plus de mémoire qu'il n'y en a réellement disponible puisque nous sommes en mode 386 étendu.

Ce phénomène est dû au ''paging''.

Attention, il ne faut pas confondre le paging avec la création des fichiers temporaires par les applications dans un répertoire dédié à cet effet et qui est paramétré par la variable TEMP placée dans le string d'environnement du DOS (commande SET TEMP = ...).

La mémoire virtuelle

Le paging, terme anglais souvent traduit par ''pagination'', désigne la technique d'utilisation d'un espace de stockage secondaire comme mémoire d'extension.

Grâce au paging, la mémoire disponible se trouve considérablement étendue au prix de temps de réponse très nettement inférieurs.

LE CONTROLEUR DU logi CIEL

ARGOS PRO

UN NOUVEAU CONCEPT DANS LE DOMAINE DE LA PROTECTION DE LOGICIELS

- Notre expérience dans le domaine de la protection, notre pouvoir d'innovation, nous permettent de vous proposer le seul produit Français utilisant des composants du type ASIC, pour une clé gigogne.
- Avantages: Circuit intégré haute densité exclusivement fabriqué pour Electryon, permettant une clé de dimension réduite, une fiabilité au fonctionnement, une transparence parfaite et une protection absolue.
- Seule clé dotée d'un numéro privé, elle permet d'identifier un PC par rapport à un autre.
- Code éditeur confidentiel sur 48 bits dont 16 programmables par vous-même - 32 à plus de 200 registres de 16 bits disponibles en lecture et écriture.

ENCORE PLUS PERFORMANTE

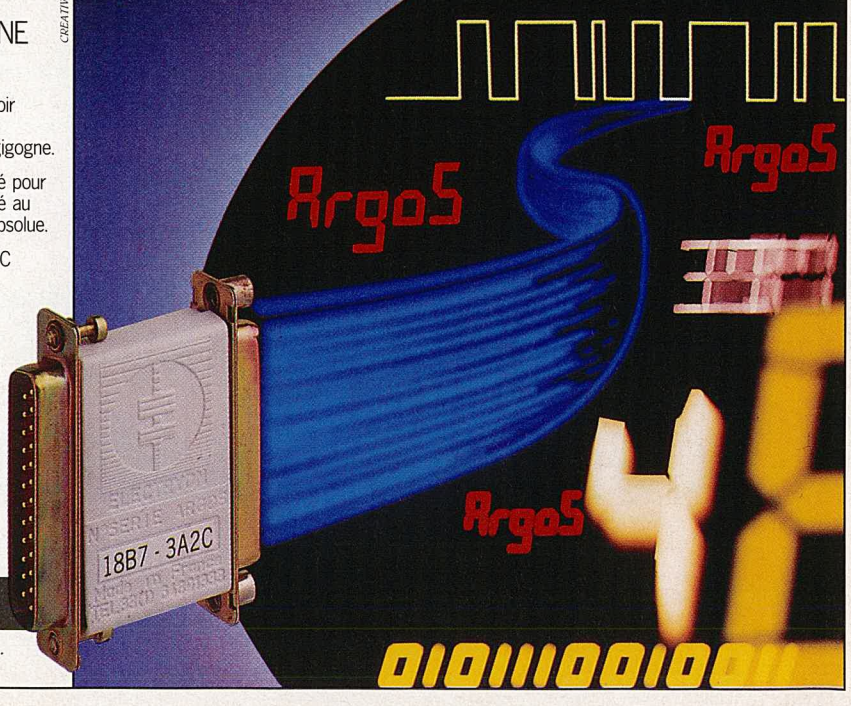
- Protection de 20 sociétés parmi 4000. Chacune de ces sociétés peut protéger plusieurs logiciels
- Activation et désactivation de la clé par programme.
- Garantie : de 1 à 3 ans selon modèle.

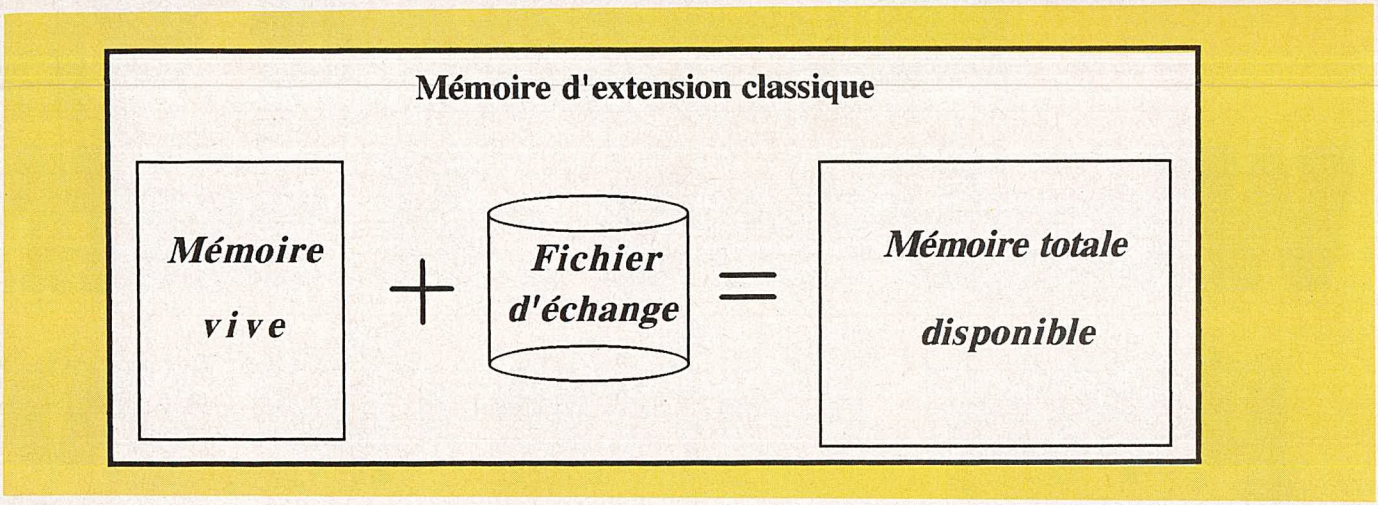
DEMANDEZ NOTRE DOCUMENTATION

ELECTRYON

53, rue Corot · La Rochette · 77000 MELUN · FRANCE.
Tél: 33(1)64 39 13 33 · Téléfax: 33(1)64 39 17 81

CREATIVE AUXERRE





Si nous disposons d'un "fichier de swap" de 2 MO (les francophones acharnés utilisent la terminologie "fichier d'échange") et que la mémoire vive de notre machine est de 3 MO, nous sommes alors à la tête d'une mémoire totale utilisable de 5 MO.

Dans ce premier cas, la mémoire supplémentaire apportée par le périphérique de stockage secondaire est simplement aboutée à la mémoire vive, donnant aux applications l'impression qu'elles disposent de plus de mémoire vive qu'il n'y en a réellement.

Mais, en mode 386 étendu, du fait des fonctionnalités propres au 80386, il y a possibilité de faire mieux encore.

Si l'on considère que l'espace mémoire occupé par une application peut être morcelé en éléments de faible taille que l'on peut à tout moment échanger avec le disque, il est alors possible de donner aux différents programmes qui s'exécutent l'illusion qu'ils disposent en permanence de la totalité de l'espace mémoire.

Lors d'un accès à une zone de mémoire, le noyau de WINDOWS sait si la parcelle correspondante est située en mémoire vive ou en mémoire d'extension et peut déterminer ainsi de son éventuel rapatriement en mémoire vive contre relégation d'une autre parcelle non nécessaire dans l'immédiat.

Voici la mémoire virtuelle telle que WINDOWS la gère en mode 386 étendu en vous donnant l'impression que vous disposiez d'environ 3 à 4 fois de la taille de la mémoire physiquement installée dans votre machine.

Il est bien évident que la qualité de la périphérie a des répercussions immédiates sur les performances générales du système : les temps d'accès au disque - périphérique privilégié d'installation des extensions mémoire - sont en première ligne de front.

Même avec des disques très rapides, il faut savoir que l'on travaille ici en millièmes de seconde (10^{-3}) alors que la mémoire réagit en milliardèmes de seconde (10^{-9}), c'est-à-dire énormément plus vite.

Heureusement, cette perte de temps n'est effective que lorsqu'il y a échange d'informations entre disque (périphérique travaillant par blocs de caractères) et mémoire vive (travaillant élément par élément) : à l'évidence, les algorithmes de détermination de paging doivent être d'une efficacité maximale sous peine de voir s'effondrer les temps de réponse de façon dramatique.

Bien entendu, WINDOWS utilise l'espace des disques pour rallonger la mémoire vive de nos machines : voyons comment.

Le fichier temporaire d'échange

En mode 386 étendu, WINDOWS travaille par défaut avec un fichier de paging temporaire qui, comme son nom l'indique, sera créé en début de session puis détruit lors de la sortie de l'environnement.

La dimension de ce fichier est automatiquement déterminée en fonction de la taille disponible sur disque. Il est cependant possible d'intervenir sur un certain nombre de paramètres définissant la localisation et la taille du fichier de relégation temporaire.

Pour cela, il faut modifier le fichier SYSTEM.INI qui sert à définir les paramètres d'exploitation de la machine. La documentation MICROSOFT conseille d'ailleurs la prudence pour toute manipulation de ce fichier, une entrée erronée pouvant empêcher WINDOWS de démarrer.

Le fichier SYSTEM.INI, comme fichier .INI utilisé par WINDOWS est un fichier de type ASCII sur lequel peut donc intervenir tout éditeur classique comme le laborieux EDLIN dont nous gratifient généreusement toutes les versions du DOS depuis l'origine.

A l'intérieur de ce fichier, les informations sont classées en rubriques dont les noms, standardisés, sont encadrés par les délimiteurs [et].

En l'occurrence, la rubrique qui nous intéresse s'intitule [386ENH].

Chaque rubrique contient un nombre variable d'entrées qui se présentent toutes sous la forme :

ENTREE = ⟨paramètre⟩

ou ENTREE désigne un mot clé suivi du signe égal puis d'un paramètre facultatif pouvant être de type logique, numérique ou n'importe quoi d'autre.

Pour intervenir sur les paramètres du fichier temporaire d'échange, il nous faut jongler avec les entrées :

- Paging,
- PagingDrive,
- MaxPagingFileSize,
- MinUserDiskSpace.

Paging suivi d'une valeur logique (YES ou NO) détermine la mise en œuvre de la fonctionnalité sur disque et donc d'extension de la mémoire vive.

En plaçant la valeur de cette entrée à NO, WINDOWS ne peut utiliser que la mémoire physiquement installée et bénéficie donc de performances nettement améliorées.

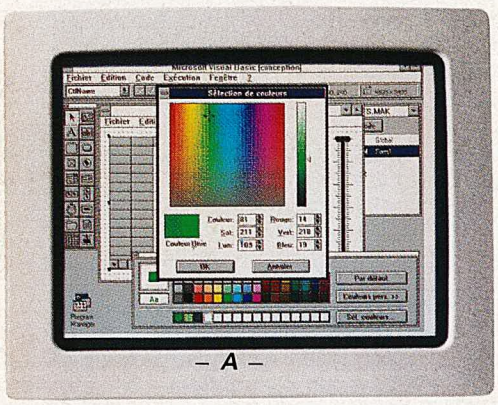
Sans mémoire virtuelle, le mode 386 étendu ressemble étrangement au mode standard si ce n'est que les applications DOS peuvent désormais fonctionner en arrière-plan.

Par défaut, l'option Paging est active, même si non explicitement précisée.

PagingDrive, qui n'est consultée que si le paging est validé, permet de choisir le disque sur lequel la pagination sera effectuée. En choisissant un disque rapide, on obtient bien sûr de meilleurs résultats, mais il ne faut pas croire qu'un disque virtuel placé en mémoire vive est la solution miracle car, en amputant la mémoire, on risque de provoquer un paging ininterrompu.

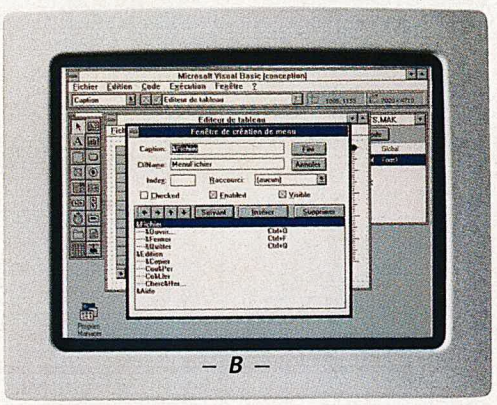
Microsoft Visu
Windows : il suffit
pour faire de voti
en programmatio

Avec Visual Basic 1.0, le monde de la programmation sous Windows ne sera plus jamais comme avant. Ce qui nécessitait parfois des mois de travail va maintenant se compter en journées.



— A —

Que vous soyez développeur ou que vous souhaitiez vous lancer dans la programmation sous Windows pour vos besoins spécifiques, Microsoft Visual Basic pour Windows est, de loin, l'outil le plus facile et le plus rapide d'emploi. Conséquence, c'est le plus productif.



— B —

***A.* Dessinez votre interface avec la souris.** Avec Visual Basic, vous créez l'interface utilisateur de votre application graphiquement sans écrire une seule ligne de code. Vous dessinez vos fenêtres en sélectionnant l'un des 15 types de contrôles disponibles dans la boîte à outils de Visual Basic. Pour les rendre plus attrayantes, vous pouvez y incorporer facilement des images que vous choisissez dans une bibliothèque de plus de 400 icônes fournie avec Visual Basic. Un éditeur d'icônes vous permet même de les modifier ou d'en créer de nouvelles.

***B.* Créez vos menus visuellement.** Vous pouvez créer vos menus très facilement et associer chaque fonction à un accélérateur clavier grâce à la fenêtre de création de menus.

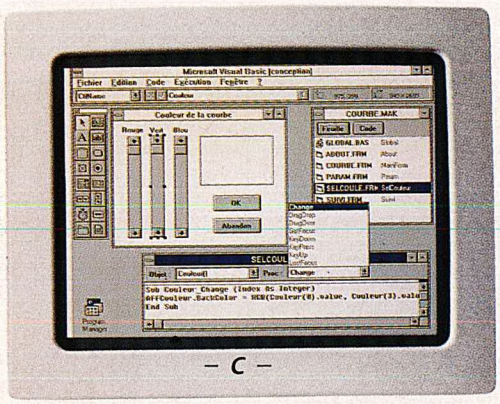
***C.* Programmez à l'aide du langage le plus accessible de la micro-informatique.** La programmation est ensuite réservée aux traitements internes de vos applications. Elle s'effectue dans un langage Basic puissant et structuré, issu de QuickBasic

★ Roux Seguela Cayzac & Goudard

al Basic pour de quelques clics e micro un crack n sous Windows.

et doté d'extensions orientées objets pour accéder facilement aux fonctionnalités de Windows.

Vous saisissez votre code directement sous Windows en le rattachant très simplement aux éléments de votre interface graphique.



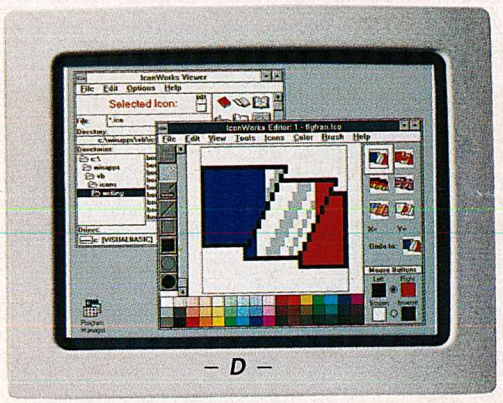
– C –

***D.* Exécutez et mettez au point votre application instantanément sous Windows.** La mise au point de vos programmes s'effectue directement sous Windows avec la possibilité d'exécuter le code pas à pas et de visualiser des variables.

Vous pouvez même modifier votre code pendant son exécution et relancer l'exécution au point d'arrêt ! Une fois la mise au point terminée, vous créez de véritables applications exécutables Windows qui sont diffusables sans run-time.

Accédez facilement à toute la puissance de Windows. Visual Basic est la plate-forme d'intégration idéale : la gestion du presse-papier, de l'impression, du lancement d'autres applications et l'échange dynamique de données (DDE) se programment très simplement.

Vous pouvez également compléter votre application avec des bibliothèques dynamiques (DLLs) fournies par des éditeurs ou développées à l'aide de langages comme Microsoft C PDS 6.0, Microsoft FORTRAN PDS 5.1, Microsoft COBOL PDS 4.5 ou Microsoft Macro-Assembler PDS 6.0.



– D –

Dans toutes les situations, vous pourrez construire votre solution. Microsoft Visual Basic est disponible en version française au prix de 1990 F HT.*

Pour toutes informations complémentaires, appelez le Service Clients au (1) 69 86 46 46 ou tapez 3616 code Microsoft.

Microsoft, 91957 Les Ulis cedex et à Infomart La Défense - Stand N° 276 niveau 2.

Microsoft®

LES LOGICIELS QUI DONNENT DES AILES.

*soit 2 360,14 F TTC prix public conseillé au 1/09/91.

Placer le fichier d'échange temporaire sur disque virtuel se révèle être une bonne solution quand :

- il y a suffisamment de mémoire vive,
- des disparités de temps d'accès à la mémoire vive placée sur carte mère et carte d'extension provoquent des plantages de WINDOWS (voir les 2 précédents articles consacrés à WINDOWS, La Revue de l'Utilisateur PC n° 71, Juin-Juillet 1991 et le n° 73 Octobre 1991).

MaxPagingFileSize détermine la taille maximale prise par le fichier d'échange temporaire. Il est ainsi possible de limiter la taille du fichier quand, par exemple, le disque récepteur ne dispose que d'un espace limité.

Il faut cependant être prudent car en-dessous d'une certaine taille du disque temporaire d'échange, WINDOWS ne fournira pas de mémoire virtuelle mais simplement une extension à la mémoire disponible (voir explications plus haut).

De façon empirique car rien n'est documenté à ce sujet, j'ai remarqué que la mémoire virtuelle n'entrait en ligne de compte que lorsqu'il y avait au moins 2 MO de fichier temporaire.

D'autre part, rien ne permet de penser que le fait de dimensionner largement le fichier d'échange temporaire va accroître les capacités de travail. En effet, pour qu'une application puisse se lancer, il faut que 2 conditions soient réunies :

- la mémoire disponible doit être suffisante,
- des "ressources système" doivent être disponibles.

Or, il n'y a, à ma connaissance, aucun moyen officiel de modifier les ressources système de WINDOWS si ce n'est par suppression des groupes de programmes, des fontes,... ; bref en ne laissant qu'un environnement dépuillé de tout ce qui fait sa richesse.

J'ai donc déjà rencontré des progiciels qui ne pouvaient plus démarrer par manque de ressources internes à WINDOWS et non par manque de mémoire.

MinUserDiskSpace impose à WINDOWS de laisser un espace libre sur le disque réservé à la relégation au moins égal à la taille du paramètre.

Cette entrée est particulièrement utile si vous utilisez un disque virtuel de faible dimension (à noter que le RAMDRIVE de MICROSOFT ne sait pas créer de disque virtuel de plus de 4 MO : d'autres utilitaires concurrents lèvent cette contrainte) et que vous souhaitez y réserver de la place pour loger des fichiers d'impression par exemple.

Par défaut, WINDOWS laissera 500 KO d'espaces libres sur le disque de relégation.

Avec ces 4 paramètres il est donc possible de placer et de "dimensionner" le fichier temporaire d'échange qui permet de mettre en œuvre la gestion de la mémoire virtuelle sous WINDOWS.

Il existe pourtant une seconde méthode qui donne d'excellents résultats : la mise d'un fichier d'échange permanent.

Le fichier permanent d'échange

Nous retrouvons ici le même principe de fonctionnement qu'avec le fichier temporaire d'échange étudié plus haut.

La différence se situe dans l'installation et le paramètrage du fichier car il n'y a plus, dans ce cas, aucune obligation de modifier le fichier SYSTEM.INI.

Bizarrement, pour créer un fichier permanent d'échange - c'est-à-dire installé à demeure sur votre disque dur - utilisable par le seul mode 386 étendu de WINDOWS, il faut passer par l'utilitaire SWAPFILE.EXE qui ne fonctionne que dans le mode réel dudit WINDOWS.

SWAPFILE.EXE qui se lance depuis le Gestionnaire de Fichier ou bien par la rubrique Exécuter du menu Fichier du Gestionnaire de Programmes, vous permet de choisir le disque sur lequel le fichier de relégation sera créé.

Vous pouvez également paramétrer sa taille que l'utilitaire propose par défaut à environ 4 ou 8 MO selon l'espace disponible sur le disque choisi.

Une fois ces paramètres définis, il suffit de cliquer sur l'option Créer pour que le fichier permanent d'échange soit mis en place de façon convenable.

Attention, des utilitaires comme PCTOOLS permettent de visualiser la présence des fichiers cachés SPART.PAR (répertoire WINDOWS) et 386 SPART.PAR (unité choisie à la création) qui correspondent à la mise en place du fichier permanent d'échange, mais il est conseillé de ne pas les supprimer à l'aide de ces utilitaires.

La façon officielle de désinstaller le fichier permanent passe à nouveau par le programme SWAPFILE, sinon, le prochain lancement de WINDOWS risque de se voir opposer une fin de non-recevoir.

Le fichier permanent d'échange est utilisé exactement comme la mémoire d'extension que nous avons étudiée au début de cet article : la taille du fichier est tout simplement additionnée à la taille de la mémoire physiquement disponible pour constituer l'ensemble utilisable.

La différence avec le fichier temporaire d'échange se situe dans la rapidité d'accès aux informations puisque SWAPFILE réserve un emplacement disque d'un seul tenant et donc non éparpillé sur l'ensemble du média récepteur commen peut l'être le fichier temporaire qui est créé par appel aux fonctions classiques du DOS. De plus, les informations contenues dans le fichier permanent sont accédées directement sans passer par les routines classiques.

Les performances générales

Travailler avec de la mémoire virtuelle est générateur d'une baisse des performances de l'ensemble puisque WINDOWS doit consacrer une partie de son temps à l'échange des informations entre disque et mémoire vive.

C'est pourquoi, la procédure d'installation initiale de WINDOWS se charge d'implanter l'ordre de chargement d'un cache-disque (SMARTDRV.SYS) dans le fichier CONFIG.SYS de la machine hôte.

Le cache-disque a pour mission de conserver en mémoire vive une copie des dernières informations échangées avec le disque. La consultation de la mémoire vive s'avérera généralement beaucoup plus rapide que l'accès au disque lui-même, améliorant du même coup les performances générales des programmes dont WINDOWS.

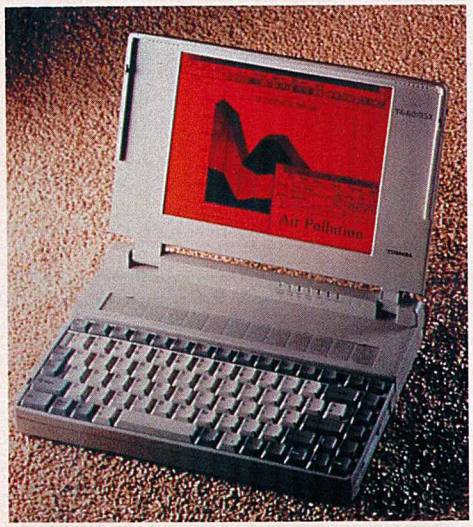
Le SMARTDRV.SYS fourni par MICROSOFT conserve en mémoire un nombre entier de pistes du disque dur. Cette méthode trouve un rendement optimal lorsque les fichiers sont rangés sur disque dans un agencement parfait, c'est-à-dire non morcelés.

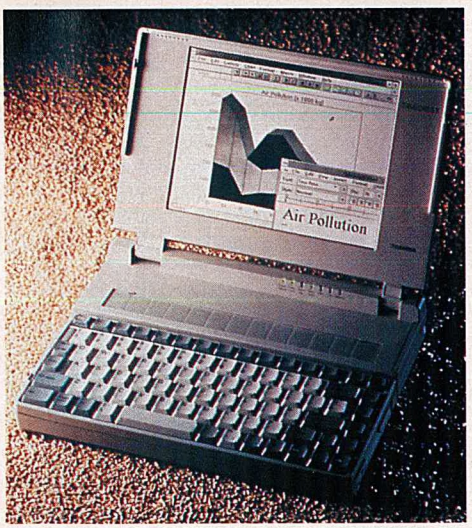
La documentation de WINDOWS nous invite donc à compacter nos disques le plus souvent possible à l'aide d'utilitaires malheureusement non fournis, mais disponibles moyennant dépense supplémentaire auprès d'autres éditeurs de logiciels.

Cette même documentation nous explique également qu'il n'est pas forcément utile de déclarer un cache-disque de grande taille car, passé certains seuils, le gain en performances devient dérisoire par rapport à la consommation mémoire générée.

LA DÉDICACE INFORMATIQUE VOUS PRÉSENTE LES DERNIERS-NÉS DE TOSHIBA EN DÉTAIL

Les nouveaux notebooks Toshiba profitent des derniers progrès réalisés en matière de miniaturisation : format A4, coque en fibre de carbone, lecteur de disquettes ultra-plat 720 ko et 1,44 Mo, batteries nouvelle génération, etc. Enfin, ils sont extractibles, c'est-à-dire qu'ils s'encastrent dans la Desk Station IV aussi facilement qu'un autoradio. Celle-ci transforme les notebooks en micros de bureau complet avec écran couleur, clavier 102 touches et deux emplacements pour cartes longues.





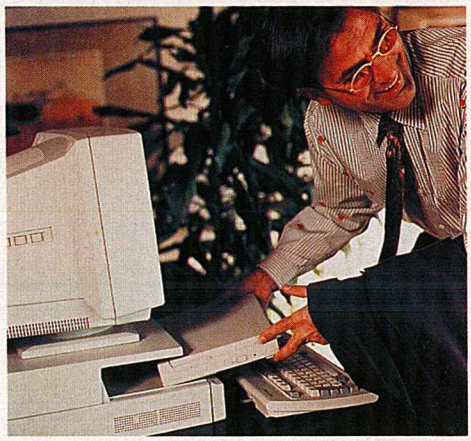
Deux versions du T4400 SX sont disponibles : une avec un écran VGA plasma et une avec un écran VGA LCD.

▲ Le T4400 SX
- Intel486 SX à 25 MHz, upgradable en Intel486 DX.
- 8 ko de mémoire cache.
- De 2 à 10 Mo de RAM.
- Ecran VGA plasma ou LCD.
- Disque dur de 80 Mo.
- 3,3 kg.

Compouce Quadri
- Modem externe de la taille d'un briquet (poids : 90 g).
- Permet l'accès aux services vidéotex (Minitel, messageries d'entreprises, etc.), le transfert de fichiers (2400 bit/s) et l'émulation des sites centraux et des serveurs utilisant la norme VT100.
- Compatible Hayes et donc avec la plupart des logiciels du marché.

Le T3300 SL
- Intel386 SL à 25 MHz.
- 64 ko de mémoire cache.
- De 2 à 10 Mo de RAM.
- Ecran LCD VGA.
- Disque dur de 80 Mo.
- Emplacement 16 bits acceptant les cartes d'extension au format cartes de crédit (modem, réseau, mémoires flash, etc.) au standard PCMCIA 2.0.
- 2,7 kg.
- Disponible fin janvier 1992. ▼

Des notebooks de la taille d'une feuille A4!



Pratique! Les nouveaux notebooks Toshiba s'encastrent facilement dans la Desk Station IV.

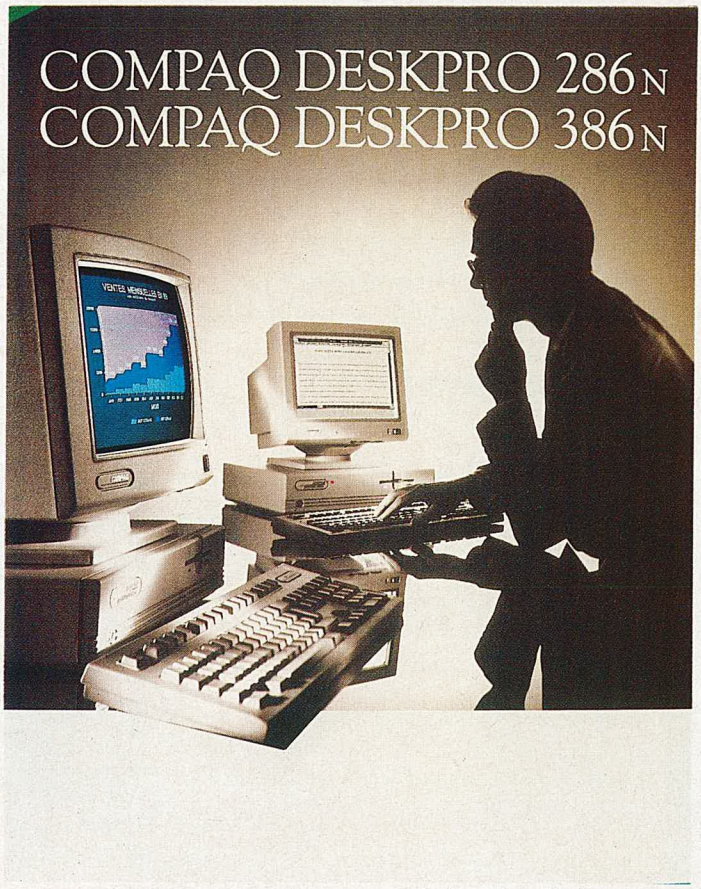
▲ Le T2200 SX
- Intel386 SX à 20 MHz.
- De 2 à 10 Mo de RAM.
- Ecran LCD VGA.
- Disque dur de 40 ou 60 Mo.
- Emplacement dédié modem au format carte de crédit.
- 2,5 kg.

L'ExpressWriter 201
- La plus petite imprimante portative du monde (303 x 80 x 44 mm), compatible avec l'ensemble des logiciels du marché.
- Ultra-légère, 850 g sans la batterie.
- Autonome (une heure avec sa batterie).
- Possibilité d'imprimer sur papier thermique, sur papier standard, transparent ou même sur des enveloppes.
- Deux polices résidentes : Courier et Prestige Elite.

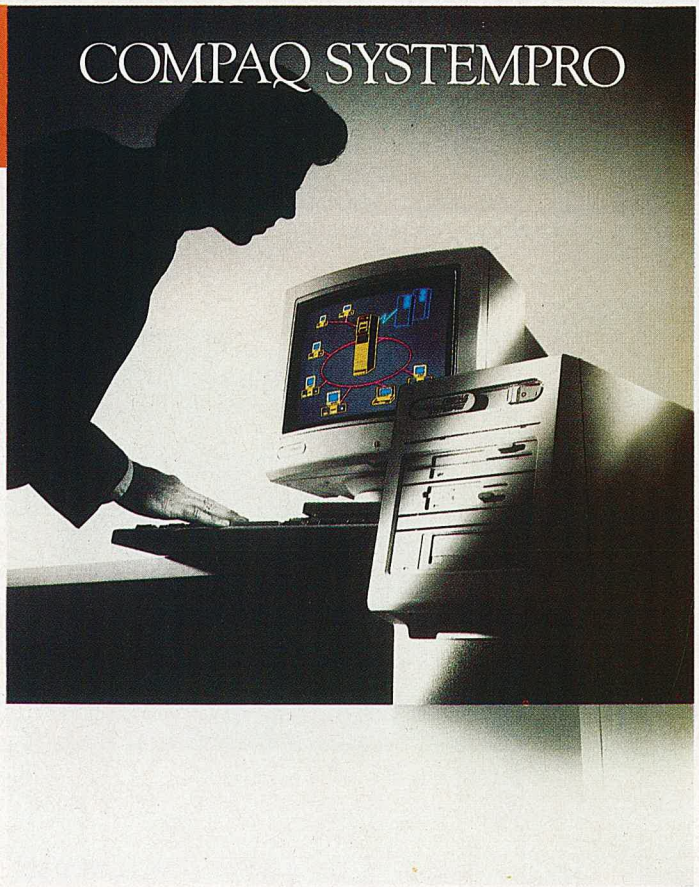
La Dédicace INFORMATIQUE

173, rue de Vaugirard 75015 Paris. Tél : 45 66 94 69. Fax : 43 06 35 18. Télex : 202 993

Equipés respectivement des microprocesseurs 286 à 12 MHz et 386SX à 16 MHz, le **COMPAQ DESKPRO 286N** et le **COMPAQ DESKPRO 386N** offrent en standard toutes les fonctionnalités des micro-ordinateurs professionnels : 3 connecteurs d'extension (2 connecteurs 8/16 bits ISA et 1 connecteur dédié à la mémoire vive rapide), 1 Mo de mémoire vive extensible à 13 Mo (286N) et 16 Mo (386N), jusqu'à 120 Mo de mémoire de masse en interne, 2 emplacements de mémoire de masse, un nouveau contrôleur graphique VGA accéléré avec support du mode texte 132 colonnes.

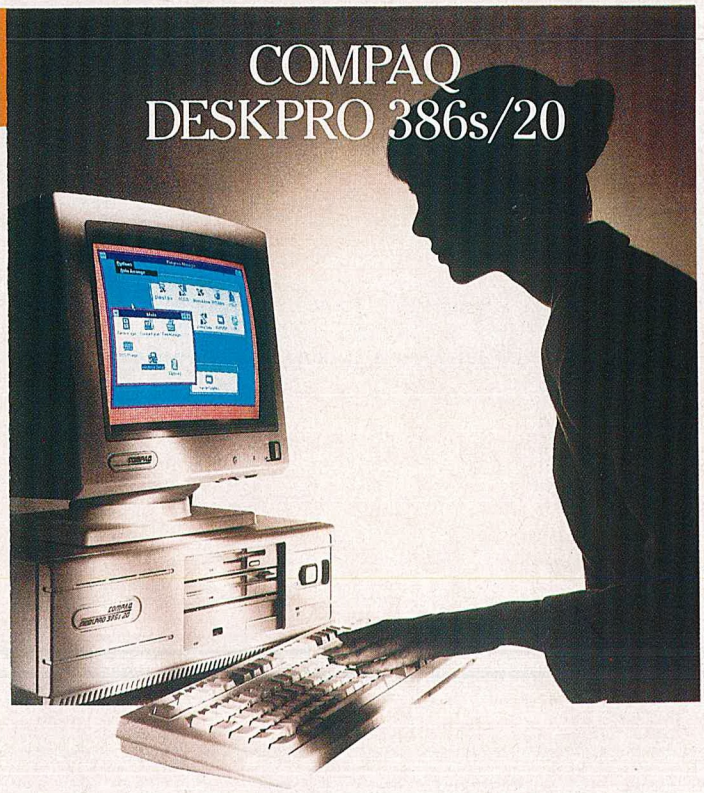


COMPAQ DESKPRO 286 N
COMPAQ DESKPRO 386 N



Basé sur une architecture évolutive en termes de processeur et de mémoire de masse, le **COMPAQ SYSTEMPRO** a été le premier micro-ordinateur à offrir un mode multi-processeur et un système multidisque. Il évolue ainsi avec les besoins des utilisateurs et préserve leurs investissements. Cette nouvelle génération d'ordinateurs a connu un grand succès : de nombreuses entreprises sont en train d'adopter le **COMPAQ SYSTEMPRO** pour faire évoluer leurs sites centraux et leur mini-informatique vers une informatique ouverte et au standard de l'industrie.

COMPAQ SYSTEMPRO

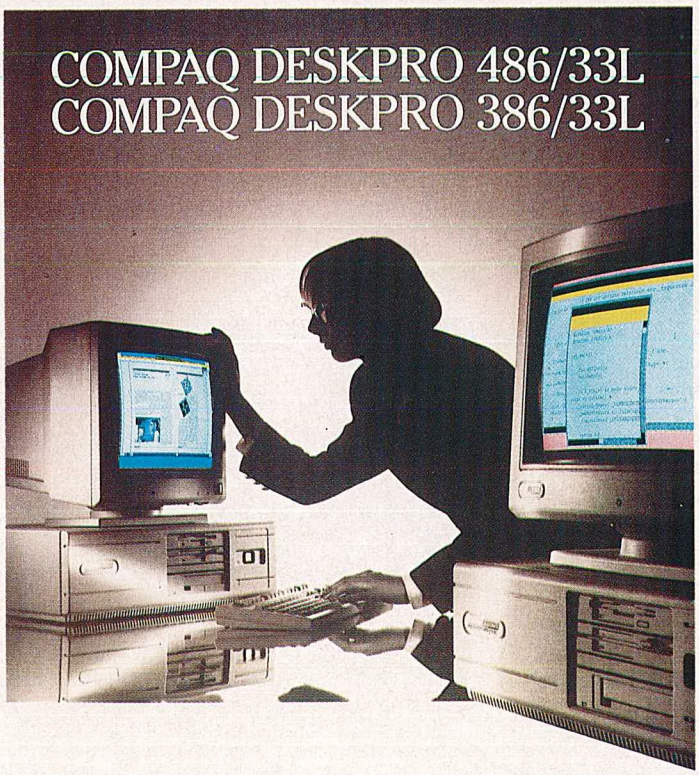


Equipé du dernier microprocesseur INTEL 80386SX à 20 MHz, le **COMPAQ DESKPRO 386s/20** est disponible en 3 modèles : 1,60 et 120 (disques durs avec temps d'accès inférieur à 19 ms). Outre sa mémoire cache ultra-rapide, le **COMPAQ DESKPRO 386s/20** est doté de nombreuses innovations. Son contrôleur VGA est ainsi accéléré pour offrir un maximum de performances en environnement graphique. De plus, il supporte le mode texte 132 colonnes utilisé par un nombre grandissant de logiciels de type tableur, traitement de texte ou d'émulation terminal.

COMPAQ DESKPRO 386s/20

Le **COMPAQ DESKPRO 486/33L** est aujourd'hui le micro-ordinateur de bureau le plus puissant du monde. En plus d'un coprocesseur 80387 et de 8 Ko de mémoire cache intégrés au processeur i486, sa carte processeur comprend un cache de deuxième niveau de 128 Ko de mémoire ultra-rapide. Ce nouveau micro-ordinateur est conçu pour des applications spécialisées techniques telles que la CAO, le développement logiciel, les SGBD et les applications scientifiques.

Le **COMPAQ DESKPRO 386/33L** dispose d'une carte processeur qui est équipée du microprocesseur 386 à 33 MHz et de 64 Ko de mémoire cache. Il constitue l'outil idéal pour les applications spécialisées telles que la CAO, les SGBD, la modélisation et l'analyse financière, ainsi que la PAO.
Un programme d'échange de la carte processeur 386 contre la carte i486 va permettre d'assurer aux utilisateurs la pérennité de leurs investissements, tout en répondant à leurs besoins d'évolutivité.



LA DEDICACE INFORMATIQUE

COMPAQ
REVENDEUR AGREE

173, rue de Vaugirard 75015 PARIS
Tél. 45.66.94.69
Fax : 43.06.35.18 - Télex 202 993

COMPAQ DESKPRO 486/33L
COMPAQ DESKPRO 386/33L

Création Parenthèse - Paris - 43 55 76 20

Se pose alors la question suivante : comment trouver le juste équilibre dans la répartition de la mémoire vive entre espace de travail et cache-disque pour obtenir des performances optimales ?

Ne comptez pas sur moi pour vous donner la réponse car je pense qu'il n'y a pas de recette universelle : chaque ''compromis'' de répartition est fonction de vos exigences en matière de performances et surtout de vos ressources machines (taille mémoire disponible, vitesse du disque, espace disque disponible,...).

S'il se trouve parmi nos lecteurs un mathématicien suffisamment émérite pour se pencher sur la question et résoudre l'équation, les colonnes de notre magazine lui sont ouvertes.

Conclusion

L'optimisation d'une configuration opérationnelle sous WINDOWS 3.0 nécessite beaucoup plus que la simple lecture de la documentation reliée du logiciel.

Une grande partie des informations que je vous ai livrées dans cet article repose sur la facture approfondie des fichiers SYSINI?.TXT que la procédure d'installation copie dans le répertoire de WINDOWS.

Le reste provient de longues heures passées à l'observation des différents phénomènes : il s'agit là d'une technique empirique dont malheureusement MICROSOFT ne nous dispense pas.

Comme au bon vieux temps du DOS 1.XX, la micro-informatique continue à nous réserver des phases de découvertes plus ou moins aléatoires qui ne sont pas sans nous rappeler l'ère des pionniers de l'Ouest Américain.

Il est vrai qu'une grande partie des innovations en arrive tout droit.

OPÉRATION "CLIPPER"

du 1er au 31 Décembre 1991

Vente exceptionnelle de **100 CLIPPER 5.01 DE NANTUCKET** Version Française
prix public de ~~10.500,00 Frs~~ H.T. **Prix exceptionnel 5900,00 Frs H.T. l'unité.**
(+ port 100 Frs)

AQUARIUM : Journal Electronique, toute l'actualité de CLIPPER, 70 pages par mois de code, commentaires, astuces et de banc d'essai.

SUPPORT TECHNIQUE : Achetez du temps pas du contrat.

NOVLIB : 190 Fonctions CLIPPER pour réseau Novell (Version Française : **2890,00 Frs H.T.**)

CLTEXT : Un vrai traitement de texte dans votre application CLIPPER.

CLREPORT, BARCODE, TELEPATHY, IMAGE TOOLKIT, NANTUCKET, TOOLS 2, etc...

Séminaire et Formation : *Paris, Strasbourg, Genève, Lyon, Marseille, Montpellier, Toulouse. Nantes.*

Demandez notre catalogue.

LDQS : 5, rue Eugène Emmanuel - 06000 NICE - Tél. 93.16.09.12 - Fax 93.88.08.90

Si vous êtes déjà un utilisateur de CLIPPER de Nantucket, vous avez besoin du

Générateur de programme
ATELIER

Il fait pour vous :

☐ la génération du menu principal,
☐ la génération de sous-menus,
☐ la génération des programmes de saisie,
☐ la génération des programmes d'impression,
☐ la gestion de vos sources,
☐ le lancement du compilateur...

ATELIER

☐ incorpore votre copyright et une calculette dans vos programmes
☐ génère le code fastidieux (90 %)
☐ et vous laisse faire la partie intéressante de la programmation.

ATELIER

☐ est livré avec les sources commentées, ainsi qu'une documentation imprimable.

La version compilée n'est pas livrée sur la disquette car ATELIER est destiné aux utilisateurs de CLIPPER et nécessite ce compilateur.

Les commentaires sur ce programme ont été publiés dans La Revue de l'Utilisateur de l'IBM PC (numéro 59, Janvier-Février 1990), et la suite de la série comportera 6 articles à compter du numéro 61, Avril 1990.

ATELIER

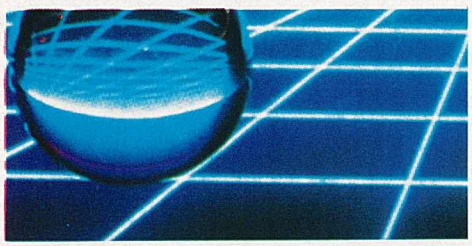
est disponible auprès de La Revue de l'Utilsateur de l'IBM PC au prix de 450 F, sur disquette 5''1/4 ou 3''1/2.

COUPON
à adresser à La Revue de l'Utilisateur de l'IBM PC
2, rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80

Nom ..... **Prénom** .....
Adresse .....
.....
Code Postal ..... **Ville** .....
Téléphone .....
Veuillez m'envoyer ..... logiciel (s) ATELIER au prix unitaire de 450 F (TTC).
Veuillez trouver ci-joint un chèque de ..... F en règlement à Softin Communications.

Une librairie de compression de données pour C, Clipper, Pascal, Basic, ASM The Data Compression Library

G. LEMEUNIER, OCIS

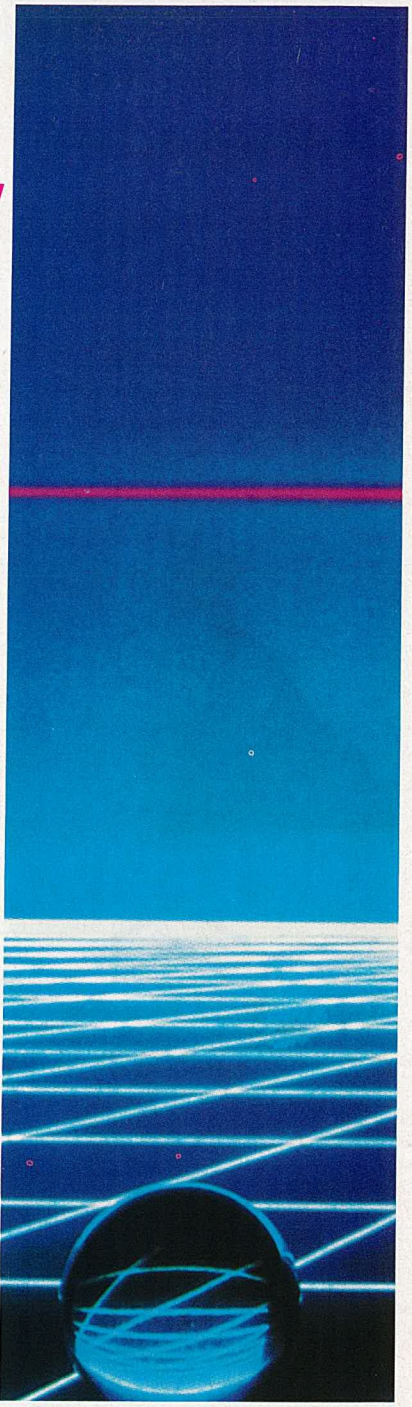


Tout le monde (ou presque) connaît le fameux utilitaire de compression de fichiers PKZIP, et combien d'entre vous auraient souhaité pouvoir intégrer un tel compacteur dans leurs applications ?

Eh bien c'est maintenant chose possible. En effet PKWARE INC. a mis sur le marché la Data Compression Library avec les cinq interfaces langages : QuickBasic, Pascal, C, Clipper, Assembleur.

OCIS à Nantes a réalisé quelques tests de cette petite merveille dont voici les résultats.

Nos tests ont porté sur les interfaces C et Clipper de la librairie.



Les fonctions du Package (I)
dans l'interface (L)
dans la librairie

```
(I) pkcopy      (dest, src, nBytes)
(I) pk_setread  ()
(I) pk_getsize  ()
(I) pk_getbuf   ()
(I) pk_setbuf   ()
(I) write_buff  (buffer, size)
(I) read_buff   (buffer, size)
(I) pk_implode  (DicSize, TypeComp)
(I) pk_explode  ()
(L) explode     ((read_buff), (write_buff), buffer)
(L) implode     ((read_buff), (write_buff), buffer, TypeComp, DicSize)
```

Les fonctions utilisateur appelées par l'interface

R__DATA() : lit un nombre x d'octets dans le fichier à compresser. Le nombre d'octets à lire est donné par pk__getsize(). La chaîne lue est transférée dans le buffer de la librairie par la fonction pk__setbuf (@ buffer, nBytes).

W__DATA() : écrit un nombre x d'octets dans le fichier de sortie. La taille du buffer de retour (il est le résultat de la compression de celui envoyé par R__DATA()) est donnée par pk__getsize, l'adresse du buffer est donnée par pk__getbuf().

Par ce système d'appel aux fonctions utilisateur, il est possible de rediriger le buffer de retour n'importe où, dans un tableau, sur la série, etc... comme nous le verrons plus loin.

Les paramètres DicSize et TypeComp de la fonction PK__IMPLODE

DICSIZE est la taille du dictionnaire que l'algorithme de compresssion va utiliser. Ce dictionnaire peut être fixé à 1 024, 2 048, 4 096 octets.

Plus sa taille est importante plus la compression est importante mais plus elle est lente, et vice-versa.

Le dictionnaire étant scanné très souvent afin d'y rechercher des séquences déjà apparues, plus celui-ci est grand, plus l'algorithme prendra de temps. Une valeur fausse provoquera le renvoi de CMP__INVALID__DICSIZE (en bref 1).

TYPECOMP est le type de compression à effectuer : 0 pour compression d'un fichier Binaire, 1 pour compression d'un fichier ASCII. La compression maximale étant obtenue avec le switch correct. La valeur CMP__INVALID__MODE (2) indique une mauvaise valeur. Dans notre source nous avons défini CMP__ASCII et CMP__BINARY.

COUIC

L'outil à désassembler

Que vous **chassiez le virus** ou que vous cherchiez à comprendre comment fonctionne un programme **COM** ou **EXE**, que vous soyez programmeur professionnel ou débutant, il vous faut un bon **désassembleur immédiat**, véritable **fenêtre ouverte** sur le code machine.

— Vous devez pouvoir **désassembler un programme** en ROM, en RAM sur disque ou sur disquette en spécifiant une adresse, **un numéro d'interruption**, un nom de fichier ou un **secteur** et faire défiler le listing de désassemblage ou de dumping en avant et en arrière.

— Vous devez disposer d'instructions **puissantes** de localisation automatique d'**interruption**, de **suivi de jump**, ou de **recherches** de chaînes et de données.

— Vous devez disposer de commandes suffisamment simples pour être apprises et manipulées sans difficulté mais assez efficaces pour faciliter la lecture du code et effectuer des **marquages** dans les zones de données.

— Vous devez pouvoir générer des **fichiers séquentiels** utilisés par les désassembleurs symboliques.

— Vous devez disposer du **code source commenté** de votre désassembleur de façon à l'adapter à vos besoins spécifiques.

COUIC-DESAS permet tout cela et plus encore grâce à ses fonctions spéciales et ses tables descriptives d'interruptions.

Exceptionnellement COUIC est vendu avec ses sources en langage "C" : plus de **9 000 lignes** de code réutilisable dans vos applications et sa documentation pour seulement **340 F (TTC)**.

COUPON
à adresser à La Revue de l'Utilisateur de l'IBM PC
2, rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80.

Nom.....

Adresse

.....

Code Postal..... **Ville**

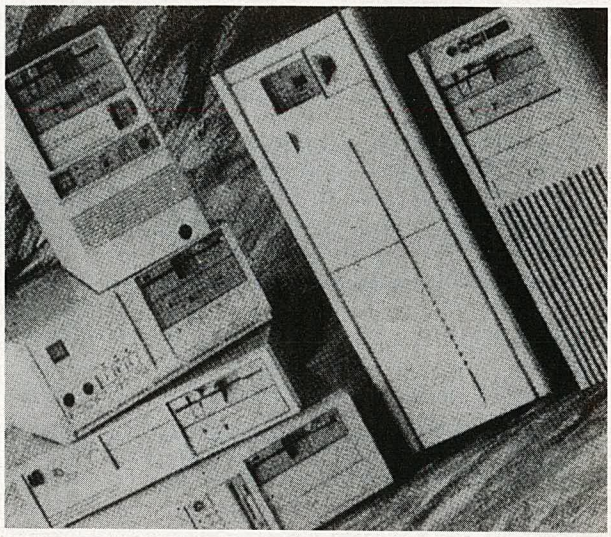
Téléphone.....

Veuillez m'envoyer ..... **logiciel(s) COUIC au prix unitaire de 340 F (TTC).**

Ci-joint un chèque de ..... **F. En règlement à Softin Communications.**

Que votre volonté soit faite !

Votre micro-ordinateur sur mesure.



Solidement implantée dans le domaine de la fabrication de micro-ordinateurs depuis plus de quatre années, **CB Informatique** vous propose une configuration adaptée à vos besoins demeurant parfaitement évolutive au fil du temps.

Testés durant 72 heures dans nos laboratoires, les micro-ordinateurs CB vous assurent de la qualité professionnelle.

Le coût ? Nous vous en laissons seuls juges !

Voici donc quelques exemples de micro-ordinateurs CB standards pour lesquels nous nous ferons un plaisir de vous proposer un devis correspondant à vos choix.

Tous nos prix s'entendent H.T. départ CB.	Hercules Monochrome	V.G.A. Monochrome	V.G.A. Couleur 800 x 600
CB 286 - 12 Disque dur 40 Mo AT Bus	**5 700**	**6 200**	**7 600**
CB 386 SX - 20 Disque dur 80 Mo AT Bus	**11 200**	**11 650**	**13 900**
CB 386 - 33 Disque dur 110 Mo AT Bus	**16 100**	**16 500**	**19 900**

Toutes ces configurations comprennent : un boîtier "Desktop", 1 mégaoctet de mémoire centrale, un lecteur 5,25"(1,2 Mo) ou 3,5" (1,44 Mo), les contrôleurs d'écran et de disques, un clavier 102 touches AZERTY ainsi que deux ports série et un port parallèle.

Tarif spécial Revendeurs : nous consulter au tél : (1) 43.96.93.18 ou Fax : (1)43.96.07.44

Je souhaite recevoir une proposition concernant le matériel suivant:

Boîtier	Carte mère	Mémoire	Lecteurs	Disque dur	Écran	Souris	MS DOS 4.01	Autres
Desktop Mini-Tower Tower	*286-12, 386 SX-20 386-20, 386-25 386-33*	*1, 2, 4, 8 Mo*	*1 Lecteur 5,25 ou 3,5 2 Lecteurs 5,25 ou 3,5*	*20,40,80,110 Mo ou supérieur*	*Hercules mono VGA mono VGA coul. 800x600*	*Oui Non*	*Oui Non*	*Par exemple : streamer, imprimante, scanner...*

Société : _____
Nom et prénom : _____
Adresse : _____
Code postal : _____
Ville : _____
Téléphone / Fax : _____ / _____

CB informatique

30, avenue du Maréchal
de Lattre de Tassigny
94410 SAINT MAURICE
Tél : (1) 43.96.93.18
Fax : (1) 43.96.07.44

Les autres valeurs de retour

```
CMP_NO_ERROR            0   Tout est ok
CMP_BAD_DATA            3   Erreur de données sur une extraction
CMP_ABORT               4   Ce code est mis à jour si la lecture du fichier à
                            décompresser renvoie 0 avant la fin.
                            (* read_buff)().
```

Liste des fichiers Make et LNK utilisées pour CLIPPER 5.01

TSTCLIP.RMK

```
//
// objets
.prg.obj:
   CLIPPER $* /M /N /B /o$@

TSTCLIP.OBJ : TSTCLIP.PRG

// executable
exemple.EXE : TSTCLIP.obj
    rtlink FI  TSTCLIP, interfce \\
           LIB implode
```

TSTCLIP.LNK (Pour BLINKER)

```
BLINKER OVERLAY FIXED
BLINKER EXECUTABLE CLIPPER F41
BLINKER INCREMENTAL ON
FILE tstclip
BEGINAREA
  FILE interfce
ENDAREA
LIB implode.lib
```

En utilisant Blinker nous rencontrons une erreur en plaçant IMPLODE.LIB en overlay, toutefois la librairie ne faisant que 8 K cela ne nous paraît pas être un handicap majeur. L'interface placée dans l'AREA ne pose aucun problème.

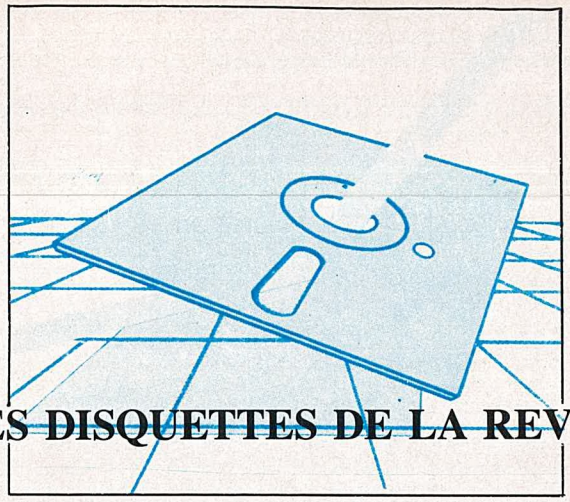
Notre source prend 3 paramètres. Cela permet d'essayer les valeurs DicSize et TypeComp et d'en apprécier les variantes. Nous avons modifié l'interface INTERFCE.C en utilisant la valeur de renvoi RetVal, dans les fonctions PK_IMPLODE et PK_EXPLODE.

La recompilation en MSC 5.10 n'a pas posé de problème. A noter qu'il faut utiliser les switchs /c /AL /FPa /Gf /Gs /Oalt /ZI pour enrayer les appels à la librairie LLIBCE.LIT au moment du link.

SOURCE CLIPPER

```
//-----------------------------------------------------------------------
//                  PROGRAMME TEST DE L'INTERFACE CLIPPER
//-----------------------------------------------------------------------

// Type de compactage
#define CMP_BINARY              0
#define CMP_ASCII               1

// Code de retour
#define CMP_NO_ERROR            0
#define CMP_INVALID_DICTSIZE    1
#define CMP_INVALID_MODE        2
#define CMP_BAD_DATA            3
#define CMP_ABORT               4
```

```
procedure main(InFile, DicSize, TypeComp)
   public inhandle      && Handle Fichier d'entrée
   public outhandle     && Handle Fichier de sortie

   ? "Test de la DATA COMPRESSION LIBRARY sous CLIPPER 5.01 (OCIS Nantes)"
   ?
   if DicSize == NIL
      DicSize := "4096"
   endif
   if TypeComp == NIL
      TypeComp := str(CMP_ASCII,1)
   endif
   if InFile == NIL
      ? "Syntaxe TSTCLIP FileName <DicSize> <TypeComp>"
   else
      COMPRESS( InFile, val(DicSize), val(TypeComp))  && Compression du fichie
      UNCOMPRESS( 'TEST.IMP', 'TEST.EXP' )            && Extraction
   endif

//---------------------- Compression du fichier InFile ----------------------
FUNCTION COMPRESS(fname, DicSize, TypeComp)

   inhandle = fopen( fname, 0 )                    && On ouvre
   if ( inhandle == -1 )                           && On plante
      ? "Erreur d'ouverture " + fname
      return .f.
   endif

   FileLen := fseek(inhandle, 0, 0)
   outhandle = fcreate( "TEST.IMP" )               && On créé le fichier de sortie

   ? time()                                        && Affiche l'heure en entrée
   ? "Compression de "+ fname + " dans TEST.IMP DicSize => " + str(DicSize);
     + " TypeComp =>"+str(TypeComp)
   && Appel à la routine C de compression avec en paramètre la taille du
   && dictionnaire et le type de compression demandé.
   RetVal = pk_implode(DicSize , TypeComp)
   ? time()                                        && Affiche l'heure en sortie
   ? "Retour de pk_implode() => "+str(RetVal)

   fclose( inhandle )                              && Fermeture des fichiers
   fclose( outhandle )
return(.t.)

//---------------------- Décompression du fichier TEST.IMP en TEST.EXP -----
FUNCTION UNCOMPRESS(inname, outname)
   inhandle = fopen( inname, 0 )                   && On ouvre
   if ( inhandle == -1 )
      ? "Erreur d'ouverture " + inname
      return .f.
   endif
   outhandle = fcreate( outname )
   ?
   ? time()                                        && Affiche l'heure en entrée
   ? "Décompression  "+ inname + " dans TEST.EXP"
   pk_explode()
   ? time()                                        && Affiche l'heure en sortie
   ? "Retour de pk_explode() => "+str(RetVal)
   fclose( inhandle )                              && Close files
   fclose( outhandle )
return .t.
```



LES DISQUETTES DE LA REVUE

DISQUETTES THÉMATIQUES		
N°	Revues	
1	10,11	Simulateur du fonct. d'un micro
2	21 à 25	Debug système et assembleur
3	17	Séquentiel indexé en BASIC
4	19 à 23	Éditeur et DAO en BASIC
5	24 à 27	Étude TURBO PASCAL (jeux, math.)
6	28	Conversion fichier DOS/WORDSTAR
7	30, 32	Étude C et PROLOG
8	30	Jeu SMILEY
9	43, 47, 49	PRSC, XPRINT et param. imprim.
10	29, 30, 33, 34	Imp. & graph., stylescript.
11	26-28, 30-37, 38	Adresses utiles + bib. souris
12	52, 53	Clavier intelligent et PASSWORD
13	40, 49	Scrolling texte + MENUDER
14	50, 51	Fenêtres Pascal + masques saisie
15	32, 34, 40	Hâchurage, Scrolling HGR, XFEN
16	48, 54	Courbes fractales
17	39, 41, 42, 43	Courbes et volumes en 3D
18	34, 35	MINITEL 1
19	36, 37	MINITEL 2
20	40, 42	MINITEL 3
21	50, 52, 53	RT/CMOS, NEWAUX, INT 44H
22	46, 48, 49	MODIF/DEPL, SAUF, BATCH inter.
23	38, 39, 41	Résidents : ZIP5, TABLES, CONTEXT
24	42, 43, 44	Assemb. & DOS (ASM, HEURE, PRSC...)
25	52, 53	Gestion de fichier en TP4
26	40, 45, 54	DEDICORE, SCAN & DEL, SAUVE
27	44, 45, 40	Étude TP. 4.0 + Fenêtrage HGR
28	48 à 54	Dossier TURBO C
29	50, 54	SECUDOS et prog. invisible
30	51, 52, 53, 55	DAO en TURBO PASCAL
31	55, 56	DBASE
32	56, 58, 59, 60...	GENECR : Générateur d'écran en T.P.
33	58, 60	Compil. interp. et désas. PASCAL

UTILITAIRES		
N°	Revues	Titres
1	40	DEDICORE : Tri sélectif des répertoires pour copier, effacer, renommer
2	41	CONTEXT : Un résident pour connaître votre configuration hard & soft
3	43	PRSC : Un résident pour rediriger et paramétrer la recopie d'écran
4	45	SCAN & DEL : Pour sélectionner et effacer vos fichiers inutiles
5	47	XPRINT : Un programme résident pour configurer l'imprimante
6	49	PARAMIMP : Un résident pour contrôler et utiliser votre imprimante
7	54	LS : Une commande DIR style UNIX
8	57	VDISK : Un disque virtuel dynamique
9	57	COPYDIR : Pour copier vos répertoires sur disquettes
10	56	RESIZE : Optimisez la taille de l'environnement de votre PC
11	58	SOSFILE : Si vous voulez récupérer des fichiers endommagés, cet utilitaire est pour vous
12	58	PCTEST : Pour tout connaître sur la configuration de votre machine
13	56, 57, 59	EXFILE : Analyseur de fichiers PASCAL
14	58, 60	HARDCOPY : Impression graphique d'écran EGA-VGA en N. & B. et nuances de gris
15	61	EDITPAL : Éditeur de palette pour EGA/VGA
16	60	XMEM : Utilisation de la mémoire étendue sous DOS
17	64	ANTIVIRUS : Traitement des virus PingPong et Jérusalem (sources et programmes)
18	70, 71, 72	XMS : OptimXMS : Optimiser la mémoire XMS

BON DE COMMANDE :
à adresser à SOFTIN COMMUNICATIONS
2 rue d'Amsterdam
75009 Paris - Tél. : 42.81.54.27

Nom .....

Adresse .....

Code Postal ..... **Ville** .....

Téléphone .....

Veuillez m'envoyer la ou les disquette (s) numéros : THÉMATIQUES (160,00 F) : .....
UTILITAIRES (90,00 F) : .....

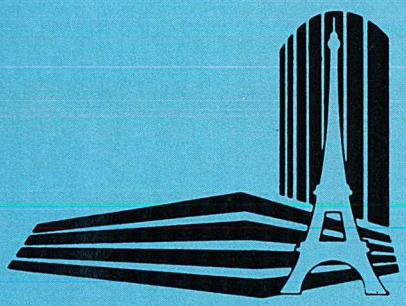
Veuillez trouver ci-joint un chèque de..... × 160,00 F soit..... Francs.
..... × 90,00 F soit..... Francs.
Total :..... Francs.

en règlement à l'ordre de Softin Communications.

Date : **Signature :**

```
//------------- Lecture des données ------------------------------------------
// Cette fonction est appelée par l'interface C c'est elle qui charge le buffer
// de données en compression ou en décompression. Elle utilise les Handles
// déclarés en PUBLIC dans la procédure Main().
FUNCTION R_DATA()
   private buff, bytesread, size
   size = pk_getsize()                          && Nombre d'octets à lire
   buff = space( size )                         && Fixe le buffer temporaire
   bytesread = fread( inhandle, @buff, size ) && Lecture
   pk_setbuf( @buff, bytesread )                && Transfert dans le buffer ZIP
return(bytesread)

//------------- Ecriture des données ------------------------------------------
// Cette fonction est appelée par l'interface C elle effectue l'écriture du
// buffer dans le fichier de sortie. Ce système assez souple permet de rediriger
// le buffer compacté vers un buffer plus grand, ou un port de communication.
FUNCTION W_DATA()
   fwrite(outhandle, pk_getbuf(), pk_getsize())
return(.t.)

 ---- SOURCE C DE L'INTERFACE CLIPPER ----------------------------------------
/*=========================================================================
=                              INTERFACE
=                   PKWARE Data Compression Library (TM)
=                                 avec
=                       Clipper Summer '87 and 5.0
=========================================================================*/


/*
Compiler en model LARGE
*/

#include "\comp\clipper5\include\extend.h"
#include "implode.h"

#define IMBUFSIZE 35256U
#define EXBUFSIZE 12574U

/* Routines utilisateur */
extern int pascal r_data( void );
extern int pascal w_data( void );

static unsigned int PK_Size;
static unsigned int PK_Dsize;
static unsigned int PK_Ctype;
static char far *PK_Buffer;

/*----- Routine de Substitution de memcpy() (question de vitesse) -----------*/
static void near pascal pkcopy(char far *dest,char far *src,int count)
{
   if (count)
   {
      do
      {
         *dest++ = *src++;
      }
      while (--count);
   }
}
```

11e Congrès de l'E.A.O. et du Multimédia



Les 7, 8 et 9 Janvier 1992
Palais des Congrès
Porte Maillot Paris 17e

Le Journal de la Formation Continue et de l'E.A.O.
2 rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80.

```
/*------------------ Indique le nombre d'octets à lire ------------------------*/
CLIPPER pk_getsize( void )
{
   _retni((unsigned) PK_Size );
}

/*---------- Renvois l'adresse du Buffer et sa longueur ---------------------*/
CLIPPER pk_getbuf( void )
{
   _retclen( PK_Buffer, (unsigned) PK_Size );
}

/*--------- Copie le buffer CLIPPER ==> Buffer PKWare -----------------------*/
CLIPPER pk_setbuf( void )
{
   if ( PCOUNT > 1 )
      PK_Size = (unsigned) _parni( 2 );
   pkcopy( PK_Buffer, _parc( 1 ), (unsigned) PK_Size );
   _ret();
}

/*------ Appelée par la fonction implode() -------------------------------*/
/* On rend ici la main au developpeur                                      */
void far pascal write_buff( char far *buffer, unsigned short far *size )
{
   PK_Size = (unsigned) *size;        /* Set Variables globales*/
   PK_Buffer = buffer;
   w_data();                          /* Appel à la fonction Clipper */
   /* Ici on peut tester la valeur de retour de W_DATA() */
}
/*------ Lecture de données même chose qu'au dessus ----------------------*/
unsigned far pascal read_buff( char far *buffer, unsigned short far *size )
{
   PK_Size = (unsigned) *size;                  /* Set Variables globales*/
   PK_Buffer = buffer;
   r_data();                                    /* Appel à la fonction Clipper */
   return(PK_Size);                             /* Return bytes read     */
   /* Ici on peut renvoyer le nombre d'octets que passe R_DATA() */
}

/*-------------- Compression ----------------------------------------------*/
CLIPPER pk_implode( void )
{
   char far *buf;
   int RetVal = 0;

   PK_Dsize = _parni( 1 );   /* Set la taille du dictionnaire */
   PK_Ctype = _parni( 2 );   /* Set le type de compression */

   buf = _exmgrab( (unsigned int) IMBUFSIZE );
   RetVal = implode( read_buff, write_buff, buf, &PK_Ctype, &PK_Dsize );
   _exmback( buf, (unsigned int) IMBUFSIZE );
   _retni(RetVal);   /* Flag d'erreur si != 0 */
}

/*-------------- Décompression --------------------------------------------*/
CLIPPER pk_explode( void )
{
   char far *buf;
   int RetVal = 0;
```

COMPUTER DISCOUNT STORE

LIQUIDATIONS • FIN DE SERIES • SAISIES • MATERIELS DE DEMONSTRATION • SOLDES...

OUVERT 7 JOURS SUR 7 (MEME LE DIMANCHE) DE 11 H A 20 H

LAPTOPS AVEC BATTERIES

- 8088/640 K/2 x 720 K — **3 990 F**
- 80286/640 K/2 x 720 K — **4 990 F**
- 80286/640 K/2 x 720 K/DD 20 Mo — **5 995 F**
- 80286/1 Mo/720 K/DD 40 Mo — **7 995 F**

MONITEURS MONOCHROMES

- 12" Bifréquence — **490 F**
- 14" Bifréquence — **690 F**
- 14" VGA — **850 F**
- Support orientable 14" — **80 F**

IMPRIMANTES NLQ MATRICIELLES

- 130 cps/80 col — **990 F**
- 160 cps/80 col — **1 290 F**
- 480 cps/80 col — **2 690 F**
- 200 cps/132 col — **1 990 F**
- 380 cps/132 col — **2 990 F**

CARTES MÈRES

- 8088 (4,77) — **295 F**
- 80286 (8) — **495 F**
- 80386/SX (16) — **2 490 F**
- 80386 (16) — **2 900 F**

LOGICIELS DE JEUX

- PRINCE — **290 F**
- F15 — **395 F**
- F19 — **395 F**
- RAILROAD TYLOON — **395 F**
- 552 — **345 F**
- GUNS — **345 F**

COMPATIBLE AT 286

1 Mo RAM • Carte + écran bimode • Clavier AZERTY
• Lecteur 1,2 Mo ou 1,44 Mo — **3 390 F**

MONITEURS COULEURS

- 12" VGA — **1 450 F**
- 14" CGA — **1 450 F**
- 14" EGA — **1 750 F**
- 14" VGA — **1 850 F**
- 14" MULTISYNC. — **2 790 F**

LASERS/FAX

- LASER 4 PPM/512 K — **5 995 F**
- LASER 8 PPM/512 K — **8 990 F**
- FAX ITC — **3 495 F**
- FAX CAFAX HS120 — **3 995 F**

GRANDES MARQUES

ONDULEURS

- 250 VA — **1 390 F**
- 350 VA — **1 690 F**

PAO/SCANNERS

- SCAN 200 DPI — **750 F**
- SCAN 400 DPI — **990 F**
- MOUSE — **190 F**
- CLAVIER TRACKBALL — **490 F**

LOGICIELS PROFESSIONNELS

- Intégré ABILITY+ (TT texte, tableur, base de données...) — **450 F**
- DAC EASY LIGHT (comptabilité) — **450 F**
- DAC EASY WORD (traitement de texte) — **450 F**

DISQUES DURS

- 20 Mo/MFM — **990 F**
- 30 Mo/RLL — **1 190 F**
- 40 Mo/IDE — **1 490 F**
- 80 Mo/IDE — **2 595 F**
- FILECARD 30 Mo — **1 595 F**

DIVERS

- Carte ECRAN CGA Hercules — **150 F**
- Clavier AZERTY — **95 F**
- Coffret Rangement 40 x 3"1/2 — **39 F**
- RAM 41256 — **125 F**
- RAM SIPP 256 K x 9 — **90 F**

DISQUETTES (par 100)

- 5"1/4 - 360 K — **1,45 F**
- 5"1/4 - 1,2 Mo — **3,50 F**
- 3"1/2 - 720 K — **3,50 F**
- 3"1/2 - 1,44 MB — **5,90 F**
- 5"1/4 couleurs — **3,25 F**

L'AFFAIRE DU MOIS

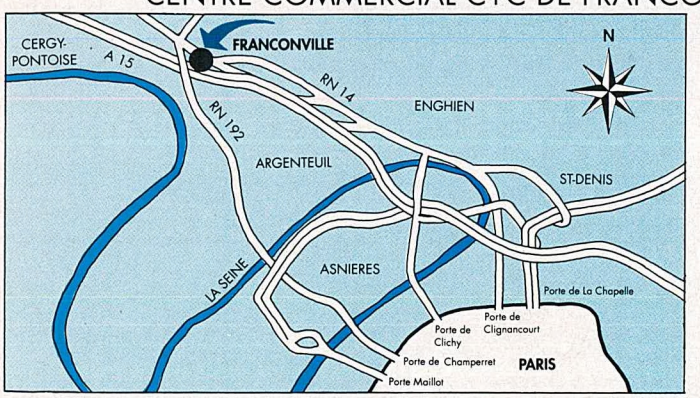
- Carte RAM + 1 Mo installés — **490 F**

Les prix sont donnés Hors Taxe, TVA de 18,6 % non comprise • Offres limitées aux stocks disponibles • Tous nos matériels sont garantis 1 mois, extension possible jusqu'à 1 an (+ 15 %) • Certains matériels peuvent présenter quelques légers défauts d'aspect (rayures, traces, petits accrocs...).

IZARD • RENNES

COMPUTER DISCOUNT STORE

CENTRE COMMERCIAL C+C DE FRANCONVILLE



Rez-de-chaussée

Accès RER : Franconville

(Plessis-Bouchard)

395, av. du Gal. Leclerc

95138 FRANCONVILLE Cedex

Tél. 30.72.55.12

Fax 30.72.55.13

LE COIN DU BRICOLEUR

Matériel présentant des défauts, défaillances, dysfonctionnement, vendu sans garantie - à réviser.

- Streamer 40 Mo Archives* — **490 F**
- Lecteur 5"1/4 / 3"1/2 — **130 F**
- Disque dur 20 Mo — **400 F**
- Moniteur 12" MONO — **250 F**
- Carte RS 232 / parallèle / horloge / jeux à partir de — **60 F**
- Cartes mères / 8088 / 80286 à partir de — **170 F**
- Programmateurs EPROM/PAL, à partir de — **350 F**
- Alimentations 180 W, à partir de — **150 F**

BON DE COMMANDE

A retourner à : COMPUTER DISCOUNT STORE
Centre commercial C+C - Lot 23 - 395, avenue du Général Leclerc - 95138 FRANCONVILLE CEDEX

Nom _____ Prénom _____

Adresse _____

Je commande les matériels suivants :

Description	P.U. H.T.	Qté	Montant net H.T.

Montant total net H.T.

TVA 18,6 %

Montant total net T.T.C.

Ajoutez 70 F de port jusqu'à 5 kg

Ci-joint mon règlement par ☐ chèque ☐ contre-remboursement

```
buf = _exmgrab( (unsigned int) EXBUFSIZE );          /* Allocation memoire */
RetVal = explode( read_buff, write_buff, buf );      /* Explode data       */
_exmback( buf, (unsigned int) EXBUFSIZE );           /* Libère la memoire  */
_retni(RetVal); /* Flag d'erreur si != 0 */
}
```

Cet outil de développement utilisé conjointement à une librairie permettant le backup et le restore. (Ex. : ClipOn.LIB) peut permettre au niveau de la sauvegarde de faire un travail relativement propre.

Quelques résultats de compression

Tests réalisés sur 386 - HDD 14 mls - CPU 33 Mhz sous Clipper 5.10

Fichier	Taille	DicSize	TypeComp	Compression	Taille	Décompression
TEST.ASC	97 k	1 024	0	5''	27 k	2''
TEST.ASC	97 k	1 024	1	5''	25 k	1''
TEST.ASC	97 k	2 048	0	7''	25 k	2''
TEST.ASC	97 k	2 048	1	6''	23 k	2''
TEST.ASC	97 k	4 096	0	7''	19 k	1''
TEST.ASC	97 k	4 096	1	7''	18 k	1''
TEST.EXE	313 k	1 024	0	19''	161 k	8''
TEST.EXE	313 k	1 024	1	19''	186 k	9''
TEST.EXE	313 k	2 048	0	22''	155 k	7''
TEST.EXE	313 k	2 048	1	23''	178 k	10''
TEST.EXE	313 k	4 096	0	30''	150 k	8''
TEST.EXE	313 k	4 096	1	30''	171 k	10''

Tests MSC 6.00

L'utilisation en C diffère de Clipper, il n'est plus nécessaire d'utiliser un module d'interface, l'accès à la librairie se faisant par des appels directs aux fonctions implode() et explode().

Le principe de passage de contrôle aux fonctions développeur pour l'envoi et la récupération d'un buffer reste toutefois le même, ce qui lui donne la même souplesse que sous Clipper.

Pour remplir et récupérer le buffer nous avons les fonctions :

ReadBuff()
WriteBuff()

Ces fonctions sont passées en paramètre à la librairie qui s'en arrange très bien. Elles peuvent stopper la compression ou la décompression en renvoyant 0. (Après un test d'espace disque ou un test série par exemple).

Bulletin d'abonnement pour 1 an

10 numéros au prix de : France : 220 F T.T.C., Etranger : 300 F
Je souscris un abonnement d'un an à la Revue de L'utilisateur de l'IBM-PC

Nom : .....
Société : .....
Adresse : .....
.....
Code postal : ..... Ville : ..... Tél. : .....

Chèque bancaire joint à l'ordre de SOFtin Communications (France).

A renvoyer à : La Revue de l'Utilisateur de l'IBM—PC, 2, rue d'Amsterdam, 75009 Paris.

POUR UNE EFFICACITE ET UNE RENTABILITE
DE LA FORMATION

Troisième Colloque
Qualité Totale de la Formation

Les 7-8-9 Janvier 1992
Palais des Congrès
Porte Maillot (Paris 17e)

Thème du colloque
Animé par des intervenants de haut niveau

Le défi des mutations des années 90 : la Qualité Totale de la Formation pour meilleure gestion prévisionnelle des compétences et des ressources humaines

Thème du 1er jour (le 7 Janvier 1992)
Les nouvelles démarche Qualité Formation et le renouvellement des outils de gestion de la Qualité pour les années 90 dans une perspective d'articulation avec la gestion des ressources humaines

Thème du 2e jour (le 8 Janvier 1992)
Méthodologie de mise en œuvre de la Qualité de la Formation

Thème du 3e jour (le 9 Janvier 1992)
Outils d'audit et d'évaluation pour la Qualité de la Formation

Renseignements et inscriptions
Le Journal de la Formation Continue et de l'E.A.O.
2 rue d'Amsterdam
75009 Paris
Tél. : (1) 46 64 07 57 - (1) 42 67 93 80

```
 ---- SOURCE MSC6 D'UTILISATION DE LA DATA COMPRESSION LIBRARY ----------------
/*****************************************************************************
TSTMSC6.C                    Appel de la                MODEL SMALL/LARGE
                  PKWARE Data Compression Library (TM)
                                 avec
                  Microsoft Optimizing C Compiler (R)
*****************************************************************************/

#include <stdio.h>
#include <dos.h>
#include <io.h>
#include <fcntl.h>
#include <malloc.h>
#include <memory.h>

#include "implode.h"
int InFile;
int OutFile;

/*
   Routines d'E/S pour les fonctions implode() et explode().
   Si ces routines renvoi 0 (Rien n'est lut ou écrit, les fonctions
   implode() et explode() s'arrêtent.
*/

/*-------------- Envois de DATA à implode() et explode() --------------------*/
unsigned far pascal ReadBuff(char far *buff, unsigned short int far *size)
{
   unsigned read = 0;
   _dos_read(InFile,buff,*size,&read);
   return(read);
}
/*------------- Récéption de DATA de implode() & explode --------------------*/
unsigned far pascal WriteBuff(char far *buff, unsigned short int far *size)
{
   unsigned wrote = 0;
   _dos_write(OutFile,buff,*size,&wrote);
   return(wrote);
}

/*------------- Procédure principale ----------------------------------------*/
void main(int argc, char *argv[])
{
   char far *WorkBuff;                    /* Buffer des tables de compression */
   unsigned error;
   char InpFile[15];
   struct dostime_t dtime;
   unsigned int DicSize = 4096;
   unsigned int TypeComp = 0;
```

```
/* Paramètre ligne de commande */
if (argc < 2) {
   printf("syntaxe ------------------------------------------------------\n");
   printf("TSTMSC6  Fichier à compacter <DicSize> <TypeComp>             \n");
   return;
   }
else {
   sprintf(InpFile,"%s",argv[1]);
}
if (argv[2] != 0) DicSize = atoi(argv[2]);
if (argv[3] != 0) TypeComp = atoi(argv[3]);

/*-- On ouvre le fichier d'entrée */
if (_dos_open(InpFile,O_RDONLY,&InFile) != 0)
{
   puts("Erreur d'ouverture du fichier d'entrée");
   return;
}

/*-- Création et ouverture du fichier de sortie */
_dos_creat("TEST.IMP",0,&OutFile);

/*-- Allocation du buffer de travail */
WorkBuff=(char far *)_fmalloc(35256U);
if (WorkBuff == NULL)
{
   puts("Erreur d'allocation du buffer de travail");
   return;
}

/*-- Compression du fichier */
printf("%s","----Compression du fichier => TEST.IMP");
printf(" DicSize: %d TypeComp: %d\n",DicSize, TypeComp);
_dos_gettime(&dtime);
printf( "%u:%02u:%02u\n", dtime.hour, dtime.minute, dtime.second );
error = implode(ReadBuff ,WriteBuff ,WorkBuff ,&TypeComp ,&DicSize);
_dos_gettime(&dtime);
printf( "%u:%02u:%02u\n", dtime.hour, dtime.minute, dtime.second );
printf("Retour de implode() => %d\n", error);
_ffree(WorkBuff);
_dos_close(InFile);
_dos_close(OutFile);

/*-- Décompactage */
WorkBuff=(char far *)_fmalloc(12574);
if (WorkBuff == NULL)
{
   puts("Erreur d'allocation du buffer de travail");
   return;
}
_dos_open("TEST.IMP", O_RDONLY, &InFile);     /* Fichier zipé      */
_dos_creat("TEST.EXP", 0, &OutFile);         /* Fichier d'extraction */
```

```
        puts("-----Décompression du fichier => TEST.EXP");

        _dos_gettime(&dtime);
        printf( "%u:%02u:%02u\n", dtime.hour, dtime.minute, dtime.second );
        error = explode(ReadBuff, WriteBuff, WorkBuff);
        _dos_gettime(&dtime);
        printf( "%u:%02u:%02u\n", dtime.hour, dtime.minute, dtime.second );
        printf("Retour de explode() => %d\n", error);
        _ffree(WorkBuff);
        _dos_close(InFile);
        _dos_close(OutFile);
    }
```

Les résultats de compression sont les mêmes que sous Clipper. Nous n'oublions pas les développeurs ASM, Pascal, QuickBasic mais le temps nous a manqué pour réaliser les tests. Toutefois des interfaces existent aussi pour eux et il nous semble tout à fait possible de l'utiliser sous Basic 7.10.

Pour ce qui est du C nous n'avons présenté que l'utilisation en MSC 6.00 mais nous avons fait fonctionner avec succès la librairie avec le MSC 5.10.

En Turbo C 2.00 et Turbo C++ 1.00 nous avons dû remplacer les fonctions qui sont propres au C de Microsoft comme le montre la source suivante :

```
 ---- SOURCE TC 2.00 D'UTILISATION DE LA DATA COMPRESSION LIBRARY ----------
/*****************************************************************************
TSTTBC2.C                       Appel de la                          SMALL/LARGE
                    PKWARE Data Compression Library (TM)
                                  avec
                              TURBO C 2.00  (R)
*****************************************************************************/
#include <stdio.h>
#include <dos.h>
#include <time.h>
#include <io.h>
#include <fcntl.h>
#include <alloc.h>
/* #include <memory.h>  n'existe pas en Turbo C*/

#include "implode.h"

int InFile, OutFile;

/*
   Routines d'E/S pour les fonctions implode() et explode().
   Si ces routines renvoient 0 (Rien n'est lu ou écrit, les fonctions
   implode() et explode() s'arrêtent.
*/

/*-------------- Envoi de DATA à implode() et explode() --------------------*/
unsigned far pascal ReadBuff(char far *buff, unsigned short int far *size)
{
   return(_read(InFile,buff,*size));
}

/*------------- Réception de DATA de implode() & explode -------------------*/
void far pascal WriteBuff(char far *buff, unsigned short int far *size)
{
   _write(OutFile,buff, *size);
}
```

```
/*------------ Procédure principale ----------------------------------------*/
void main(int argc, char *argv[])
{
   char far *WorkBuff;                    /* Buffer des tables de compression */
   unsigned error;
   char InpFile[15];
   unsigned int DicSize = 4096;
   unsigned int TypeComp = 0;
   struct time dtime;

   /* Paramètre ligne de commande */
   if (argc < 2) {
      printf("syntaxe ------------------------------------------------------\n");
      printf("TSTMSC6  Fichier à compacter <DicSize> <TypeComp>             \n");
      return;
      }
   else {
      sprintf(InpFile,"%s",argv[1]);
   }
   if (argv[2] != 0) DicSize = atoi(argv[2]);
   if (argv[3] != 0) TypeComp = atoi(argv[3]);


   if ((InFile=_open(InpFile,O_RDONLY)) <= 0)
   {
      puts("Erreur d'ouverture du fichier d'entrée");
      return;
   }

   /*-- Création et ouverture du fichier de sortie */
   OutFile=_creat("TEST.IMP",0);

   WorkBuff=(char far *)malloc(35256U);

   /*-- Allocation du buffer de travail */
   if (WorkBuff == NULL)
   {
      puts("Erreur d'allocation du buffer de travail");
      return;
   }

   /*-- Compression du fichier */
   printf("%s","----Compression du fichier => TEST.IMP");
   printf(" DicSize: %d TypeComp: %d\n",DicSize, TypeComp);
   gettime(&dtime);
   printf( "%u:%02u:%02u\n", dtime.ti_hour, dtime.ti_min, dtime.ti_sec );
   error = implode(ReadBuff ,WriteBuff ,WorkBuff ,&TypeComp ,&DicSize);
   gettime(&dtime);
   printf( "%u:%02u:%02u\n", dtime.ti_hour, dtime.ti_min, dtime.ti_sec);
   printf("Retour de implode() => %d\n", error);
   free(WorkBuff);
   _close(InFile);
   _close(OutFile);

   /*-- Décompactage */
   WorkBuff=(char far *)malloc(12574);
   if (WorkBuff == NULL)
   {
      puts("Erreur d'allocation du buffer de travail");
      return;
   }
```

```
    InFile=_open("TEST.IMP",O_RDONLY);      /* Fichier zipé      */
    OutFile=_creat("test.ext",0);           /* Fichier d'extraction */

    puts("----Décompression du fichier => TEST.EXP");
    gettime(&dtime);
    printf( "%u:%02u:%02u\n", dtime.ti_hour, dtime.ti_min, dtime.ti_sec);
    error = explode(ReadBuff, WriteBuff, WorkBuff);
    gettime(&dtime);
    printf( "%u:%02u:%02u\n", dtime.ti_hour, dtime.ti_min, dtime.ti_sec);
    printf("Retour de explode() => %d\n", error);
    free(WorkBuff );
    _close(InFile);
    _close(OutFile);
}
```


LA DATA COMPRESSION LIBRARY

UNE SOLUTION SIMPLE ET EFFICACE POUR :

- DEPASSER VOTRE ESPACE DISQUE
- OPTIMISER VOS SAUVEGARDES
- ACCELERER VOS TRANSFERTS DE DONNEES
- LIVRER VOS SOFTS AVEC DES INSTALLATIONS DE "PRO"

COMMANDEZ-LA DES AUJOURD'HUI

BON DE COMMANDE

SOCIETE _____ RESPONSABLE _____

ADRESSE _____

VILLE _____ CODE POSTAL _____

TEL _____ FAX _____

Souhaitons recevoir la disquette "PROGRAMMES TESTS" ..... ( )
Souhaitons recevoir la DATA COMPRESSION LIBRAIRY à 3550.00 F TTC ..... ( )
Règlement par chèque à la commande

O.C.I.S. - 3 bis Place Neptune 44000 NANTES - Tél. : 40 08 05 03 - Fax. : 40 48 75 77

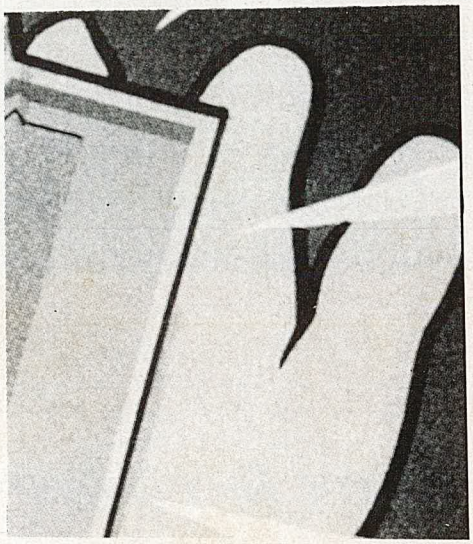
MULTIMEDIA
ET
ENSEIGNEMENT DES LANGUES

dans le cadre du XIe Congrès de l'E.A.O.
et du Multimédia

Le 7 Janvier 1992
Palais des Congrès
Porte Maillot (Paris 17e)

Renseignements et inscriptions
Le Journal de la Formation Continue et de l'E.A.O.
2 rue d'Amsterdam
75009 Paris
Tél. (1) 47.64.07.57 - (1) 42.67.93.80 - (1) 42.67.92.99

Clipper 5.01, TBrowse : les couleurs



Dominique BRUYERE

Une approche de TBrowse par son côté le plus attrayant : les possibilités d'utilisation de la couleur dans des fenêtrages attractifs.

Fallait-il aborder, dans la Revue de l'Utilisateur PC, la grosse machine qu'est TBrowse, par un article sur les possibilités qu'il nous offre de gérer les couleurs ? Je pense que c'est une façon d'aborder l'usine à gaz par son côté le plus chatoyant et le plus immédiatement visible.

Avec son approche, à pas feutrés il est vrai, dans le domaine de la Programmation Orientée Objet, Clipper 5 nous fait le cadeau d'une transition forcée mais douce. Là où tout le monde vous parle des difficultés à maîtriser les codes blocks et de la complexité des objets, j'ai choisi de vous parler, en cette fin d'été, des couleurs ; comment utiliser la couleur avec TBrowse ? et comment tirer parti du formidable outil de développement que nous offre Nantucket ?

Il est peut-être nécessaire de revisiter les nouvelles possibilités que Clipper 5 nous apporte, dans le domaine de la gestion esthétique des écrans.

1) La possibilité, liée à l'environnement matériel, de modifier la densité d'affichage : Setmode() nous permet de passer de 80 colonnes à 40, et de 25 lignes à 43 et 50 lignes.

2) Les couleurs, grâce à Setblink(.F.), passent à 256, sur écran approprié, bien sûr.

3) Les commandes SAY et BOX supportent la clause supplémentaire : ⟨COLOR⟩.

4) La commande SET COLOR TO est à éviter dorénavant, et à remplacer par SETCOLOR(), qui permet une sauvegarde de l'environnement, comme d'ailleurs toutes les commandes SET xxxxx(). On n'en finira pas de s'extasier sur ce confort supplémentaire apporté aux programmeurs. Mais ceci fera peut-être l'objet d'un futur article.

Cet article va, tout doucement, vous faire parcourir quelques-unes des possibilités gigantesques que nous apporte TBrowse.

Il faut rappeler qu'une classe, telle que livrée par Nantucket, comporte des variables d'instances exportées, assignables ou non, et des méthodes, qui nous permettent d'agir sur les objets créés par les classes.

Les méthodes les plus évidentes de TBrowse sont, bien entendu, des méthodes de déplacement du curseur dans la fenêtre. Les variables d'instances les plus courantes gèrent l'aspect technique de la fenêtre.

Rappelons que TBrowse permet de créer un objet vide, doté de particularités et de capacités, mais rigoureusement vide.

Il faudra, par la suite, le remplir avec l'aide de TBColumn, et enfin l'activer par la méthode Stabilize() qui devra renvoyer une variable d'instance : ''stable'', initialisée à .T.

Jetons un un œil sur les méthodes de TBrowse, que nous allons utiliser, ainsi que sur les variables d'instance des deux classes qui nous intéressent : TBColumn et TBrowse.

ColorSpec Variable d'instance exportée assignable. Classe : TBrowse

ColorSpec contient une chaîne de type caractère donnant le contenù de la table de couleurs pour l'affichage du TBrowse.

C'est la valeur du SETCOLOR() en cours d'utilisation, qui est copiée par défaut dans cette variable, au moment de la création de l'objet TBrowse.

ColorSpec est un tableau, mais pour d'obscures raisons, il doit être initialisé par une chaîne caractère et non par un tableau littéral, comme on pourrait s'y attendre.

Il y a donc deux façons d'intervenir sur la valeur de ColorSpec : en la modifiant directement, ou bien en modifiant la valeur de SetColor().

Le nombre de couleurs que l'on peut passer à ColorSpec paraît sans limites, à part la fatigue visuelle qui pourrait en résulter.

TBrowse utilisera, par défaut, la seconde couleur pour afficher les données situées à l'emplacement du curseur. Toutes les autres données seront affichées en utilisant la première couleur.

DefColor Variable d'instance exportée assignable. Classe : TBColumn

DefColor contient un tableau composé de deux valeurs numériques utilisées comme index dans la table des couleurs de l'objet TBrowse. La première valeur représente la couleur non sélectionnée, couleur qui est utilisée pour afficher les titres, les bas de pages, les données quand le curseur interne à l'objet n'est pas positionné sur les valeurs affichées.

La seconde valeur, vous l'aviez compris, représente la couleur sélectionnée, couleur qui, elle, a pour rôle d'afficher la cellule sur laquelle se trouve le curseur.

La valeur par défaut de TBColumn : DefColor est {1,2}. Cela fait que les deux premières couleurs présentes dans la table des couleurs de TBrowse sont utilisées respectivement comme couleur non sélectionnée et sélectionnée.

N'oubliez pas que les couleurs définies en employant TBColumn : DefColor sont annulées par une éventuelle utilisation de TBColumn : ColorBlock décrit ci-dessous.

3615 TEASER

Recevez ***GRATUITEMENT*** *le logiciel BBT pour télécharger avec votre machine* ***(PC XT/AT - ATARI ST - AMIGA - MAC)*** *et venez prendre nos logiciels du domaine public !*

3615 TEASER

Plus de ***10.000 logiciels*** *triés et sélectionnés à votre disposition. Faites votre choix parmi eux. Ils seront chez vous en quelques minutes prêts à l'emploi !*

3615 TEASER

Notre protocole BBT est un des plus rapides (90 cps) et des plus fiables du marché sous Transpac et nos logiciels sont ***les meilleurs et les plus récents.***

3615 TEASER

En quelques minutes chez vous les derniers softs pour ***PC XT/AT, ATARI ST, AMIGA*** *et* ***MAC*** *: tableurs, traitements de textes, langages, graphisme, musique, section adultes et des jeux par milliers.*

Pour recevoir votre **BBT**, adressez à :

FRANCE-TEASER

22, Grande Rue 92310 SEVRES

une disquette vierge avec votre nom, prénom, adresse et type d'ordinateur. Joignez 15 francs en timbres pour frais d'expédition. Vous le recevrez sous 48 h.

Création typon : Club Megaland - 3615 MEGALAND

ColorBlock Variable d'instance exportée assignable. Classe : TBColumn

ColorBlock contient un code block optionnel qui détermine dans quelle couleur les données vont être affichées.

S'il est présent, à chaque fois qu'une valeur est récupérée, le block est exécuté à l'aide de TBColumn : block (le bloc de récupération des données).

La nouvelle valeur est passée comme argument à TBColumn : block, qui, lui, renvoie un tableau contenant deux valeurs numériques. Ces valeurs servent alors l'index dans la table de couleurs de l'objet TBrowse, comme défini ci-dessus.

TBColumn : ColorBlock permet d'afficher les données dans une couleur prédéterminée, basée sur le contenu de ces données. Il est désormais facile d'afficher toutes les cellules d'un champ logique : expédition en vert, si l'expédition a été effectuée, en rouge dans le cas contraire. Nous en verrons des exemples plus loin.

Un conseil en passant : utilisez VALTYPE() au lieu de TYPE(), pour tester les types de données au moment de la définition de c : ColorBlock.

ColorRect (<aRect>, <aCouleurs>). Méthode. Classe : TBrowse 5.01

ColorRect() modifie directement la couleur d'un groupe rectangulaire de cellules. <Arect> est un tableau de 4 nombres (haut, gauche, bas et droite). Ces nombres font référence, comme dans un tableur, aux cellules et non aux coordonnées classiques de l'écran.

<aCouleurs> est un tableau de deux nombres. Ces nombres servent d'index dans la table des couleurs de l'affichage du Browse. Ces couleurs deviennent la couleur normale et la couleur en surintensité des cellules situées à l'intérieur du rectangle.

Les cellules colorées par ColorRect() gardent leur couleur jusqu'à ce qu'elles soient sorties de la fenêtre du Browse, soit vers le haut soit vers le bas. Un déplacement latéral n'a pas d'effet sur cette couleur. Une colonne comprise dans un rectangle garde donc sa couleur même si elle n'est pas visible dans la fenêtre.

Cet exemple colore la totalité de la fenêtre virtuelle, sans souci de ce qui peut être affiché sur l'écran et hors écran.

RowCount et ColCount sont deux variables d'instances de la classe TBrowse. RowCount donne le nombre de lignes visibles dans le Browse, ColCount donne le nombre total de colonnes dans le Browse.

LES DISQUETTES DE LA REVUE
GESDOC

Logiciel de recherche documentaire (voir page .....).
Livré sur disquette 3.5'' programme et sources utilisable directement ou modifiable par vous-même.
Permet la recherche par auteur, sujet, titre, éditeur, mots-clés,...
Remplace des logiciels qui sont vendus dix fois plus cher...
Une application générée à l'aide de ATELIER !

COUPON
à adresser à La Revue de l'Utilisateur de l'IBM PC
2, rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80.

Nom ..... **Prénom** .....
Adresse .....
.....
Code Postal ..... **Ville** .....
Téléphone .....
Veuillez m'envoyer ..... logiciel(s) GESDOC au prix unitaire de 450 F (TTC).
Ci-joint un chèque de ..... F. En règlement à Softin Communications.

nPosHaute : = // à partir de la première cellule

nPosGauche : = // la plus à gauche

aRect : = {1,1, browse:rowCount, browse:colCount}

browse: colorRect(aRect, {2,1})

Nous verrons un autre exemple plus amusant et plus utile tout à l'heure.

Hilite(). Méthode. Classe : TBrowse 5.01

Met en surintensité la cellule sur laquelle le curseur est positionné. Cette méthode est à utiliser quand la variable d'instance TBrowse : autoLite est à .F.

DeHilite(). Méthode. Classe : TBrowse 5.01

Supprime la surintensité dans la cellule sur laquelle le curseur est positionné. De même que pour la méthode Hilite(), TBrowse : autoLite doit être à .F.

Justement qu'allons-nous pouvoir faire avec nos couleurs ?

Quelques idées :

— modifier la couleur d'une cellule en fonction de son contenu ;

— modifier la couleur d'une colonne si elle est ''freezée'' ;

— marquer un bloc dans un objet TBrowse et l'exporter ;

— modifier la couleur des titres ;

— modifier la couleur d'un enregistrement s'il est ''Deleted(.T.)'', etc.

Le programme de démonstration qui suit est un tout. Il crée tout seul un fichier destiné à être utilisé tout au long de la démonstration.

TBrowse souffre d'un défaut majeur : notre manque d'imagination !

Nous commençons à balbutier avec Clipper 5, j'espère que nous atteindrons vite un niveau plus élevé. Ce n'est que par une participation effective de tous les développeurs, que ce niveau peut s'élever. La fonction du siècle n'a jamais suffi à faire vendre un programme.

Ne cherchez pas, il n'y a dans ce petit exercice que des choses simples, trop simples à mon gré, mais qui sont des exemples de ce que l'on peut faire avec TBrowse.

Cet exemple contient beaucoup de possibilités très puissantes de Clipper 5, elles sont appliquées, mais pas expliquées. La pagination réduite qui est la nôtre, nous oblige à repousser à plus tard des explications complémentaires.

Du côté des news et plus particulièrement des produits nouveaux, ou en passe de s'imposer, et pour lesquels des tests vont être nécessaires.

Les disquettes de la Revue

ARTOOL UN LOGICIEL DE D.A.O. POUR PC ET CARTES EGA OU VGA

ARTOOL possède toutes les fonctions pour exploiter pleinement les possibilités graphiques de votre PC muni d'une carte EGA ou VGA. Il fonctionne en mode BITMAP, et gère toute la palette de couleurs.

Spécifications techniques :

☐ Fonctionnement en 16 couleurs, 640×350 pour EGA et 640×480 pour VGA.

☐ Dessin à ''main levée'', tracé de droites, de cercles et de boîtes.

☐ Ajustement d'une courbe par la méthode de Bézier .

☐ Fonctions avancées : loupe, miroir, coloriage, texte.

☐ Gestion de la palette, des fichiers images et de l'imprimante.

ARTOOL fonctionne sur tous compatibles PC muni d'une souris et d'une carte graphique EGA ou VGA. Un minimum de 640 Ko ainsi qu'un disque dur sont vivement recommandés. Des impressions graphiques peuvent être obtenues sur des imprimantes compatibles IBM-GRAPH ou EPSON.

Fourni avec les sources en langage PASCAL, vous pourrez adapter ARTOOL à vos besoins et à votre configuration.

ARTOOL est disponible sur disquette 5'' 1/4 (360 Ko), au prix de 450 F (TTC)

BON DE COMMANDE
à adresser à La Revue de l'Utilisateur PC
2 rue d'Amsterdam
75009 Paris - Tél. : 42.81.54.27

Nom .....

Adresse .....

Code Postal ..... Ville ..... Téléphone .....

Veuillez m'envoyer ..... logiciel(s) ARTOOL, au prix unitaire de 450 F TTC.

Ci-joint un chèque de ..... F.

En règlement à l'ordre de Softin Communications.

CLIPPER

En premier le produit dénommé ''ALORS'', est un environnement de développement pour Clipper 5 qui a été mis au point par l'équipe de P.7.I. Développement.

Cet environnement établit une norme en matière de dictionnaire de données, générateur de source et bibliothèque. Son objectif est de permettre une maintenance aisée des applications. ''ALORS'' possède un véritable dictionnaire de données. Celui-ci contient le descriptif des divers fichiers d'une application, les contrôles élémentaires sur chaque champ, ainsi que les liens interfichiers.

Ces contrôles sont maintenus par le dictionnaire indépendamment des lignes de programme. Ceci évite de répéter l'erreur des ''boîtes à outils'' qui imposent des contrôles dans les écrans mêmes, les multipliant ainsi à travers toute l'application.

Rassurez-vous, développeurs de tout ''poil'', ''ALORS'' ne vous mettra pas sur la paille. Il vous laisse le soin d'écrire du code personnalisé, limité aux algorithmes de l'application.

Contrairement aux autres générateurs du marché, il ne reste au développeur qu'à maintenir une très petite quantité de code : celle réellement liée à l'application.

Le dictionnaire de données effectue les contrôles suivants indépendamment du source du programme :

— pictures automatiquement associés aux champs des fichiers ;

— obligation ou non de remplir le champ ;

— contrôle de saisie automatiquement déclenchés lors de la mise à jour d'un champ d'un fichier.

Les générateurs :

— le générateur de procédures crée le programme appelant et toutes les procédures nécessaires à une application ;

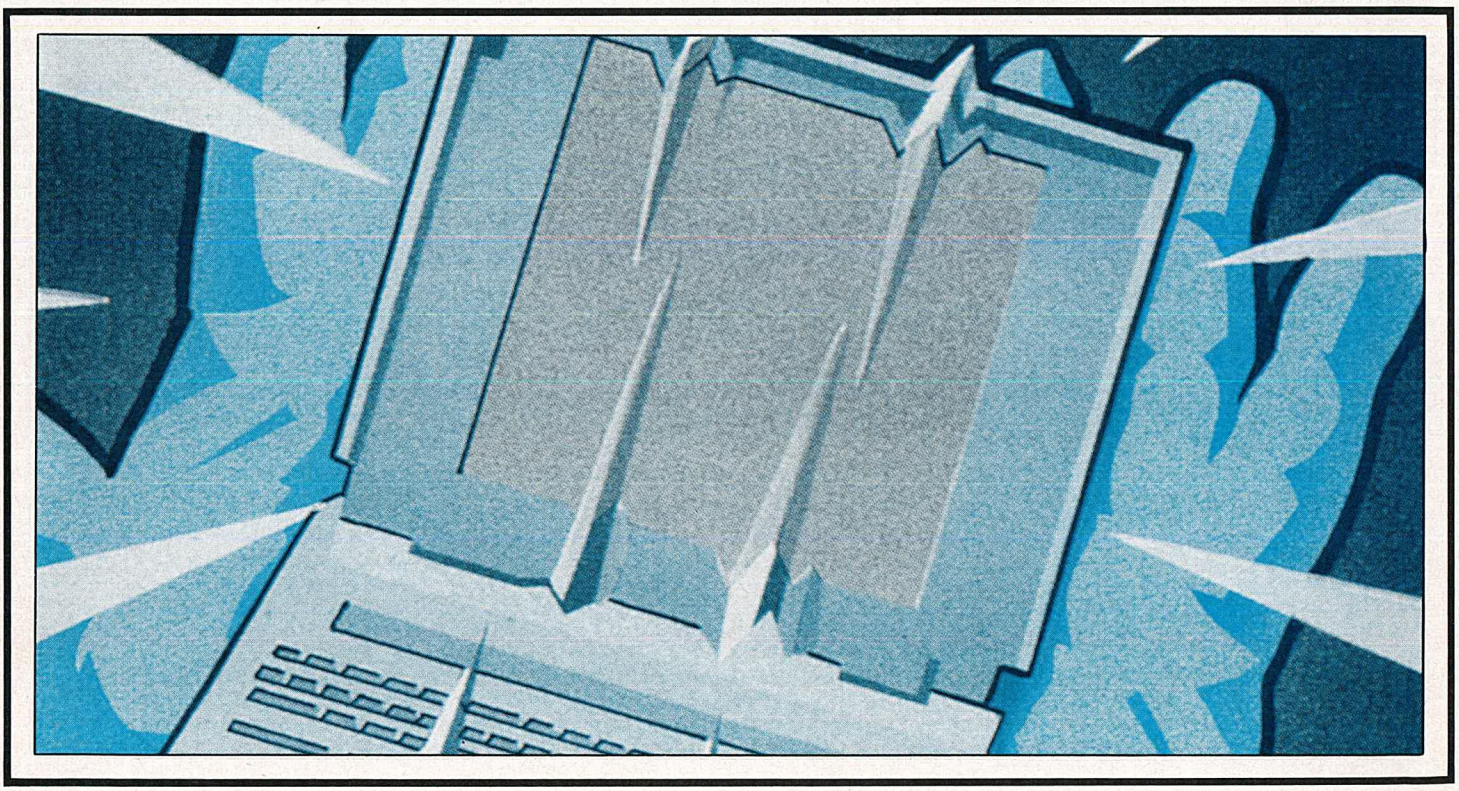
— un générateur de menus déroulants ;

— un générateur d'environnement de compilation ;

— un générateur de documentation technique (bien utile depuis que SNAP refuse de travailler avec Clipper 5).

Bien sûr, en prime, ''ALORS'' gère les couleurs, les mots de passe, les erreurs, les drivers d'imprimante, les fenêtres et la souris.

Une aide utilisateur plein écran est présente et les applications sont générées pour une utilisation en réseau.



Je développerai une application complète sous peu, avec l'aide de "ALORS", et pourrai vous en parler plus profondément.

Une version de démonstration est disponible, au prix de 150 F, chez P.7.I., 88 avenue Kléber, 75008 Paris, Tél. 16 (1) 47.27.01.29.

Avez-vous déjà rêvé de voir votre programme Clipper téléphoner à votre place ? Envoyer ou recevoir un fichier à un autre ou d'un autre ordinateur ?

Travailler avec un lecteur de code-barres ? Ecrire un programme de serveur, style BBS ? Faire votre propre programme d'émulation de terminal ? Et oui..., écrire vous-même votre version personnelle d'un programe comme AQUARIUM ?

C'est tout ce que vous apporte "TELEPATHY"

TELEPATHY est une librairie de communications pour Clipper 87 et Clipper 5.0. Avec elle, il est facile de vous sentir très à l'aise quand quelqu'un vous demande si vous êtes capable de réaliser des communications à partir d'un port série, et ce, depuis une application Clipper.

Les fonctions de "TELEPATHY" couvrent les catégories suivantes :

— Contrôle du port série.
— Statut/Contôle de l'UART.
— Réception de données.
— Envoi de données.
— Contrôle du modem.
— Protocole de transfert de fichier.
— Emulation d'un terminal.
— Contrôle des bits et des octets.
— Annonce d'un événement.
— Contrôle des erreurs.

Tous les "famas" de Clipper aimeront la façon dont ont été réalisées les fonctions utilisées dans TELEPATHY. La plupart des fonctions sont du type métafonctions. Par exemple, au lieu d'avoir une fonction pour récupérer l'actuel paramètrage de l'UART en Bauds, et une autre pour changer ce paramètrage, il n'y en a qu'une, appelée tp__baud().

Cette fonction retourne l'actuel paramètrage, et initialise une nouvelle vitesse de travail, si désirée, de la même manière que Clipper à travers sa fonction SETCOLOR() et ses fonctions SET().

De plus, les fonctions qui contrôlent l'état du modem et de l'UART, ont été faites de façon à retourner des informations sur chaque bit avec un seul appel, plutôt que d'avoir des fonctions séparées pour tester l'état de chaque registre intéressant.

Tout cela me semble très bien pensé, et tout utilisateur de Clipper devrait apprécier cette façon de voir.

Les protocoles de transfert de fichier supportés par TELEPATHY sont exceptionnels.

En voici la liste :

Xmodem

Xmodem-CRC

Xmodem-1K (quelquefois appelé Ymodem)

Ymodem (quelquefois appelé Ymodem batch)

Ymodem__G

Zmodem

Kermit

En plus, toutes les fonctions nécessaires au contrôle des paramètrages optionnels de ces protocoles sont fournies.

Une fonction paramètrable nommée Progress(), peut être spécifiée pour chacun de ces protocoles, et permet un affichage personnalisé de la progression des différents états durant le transfert de fichier.

Faire transiter de l'information par le port série est très simple.

Après avoir initialisé le buffer de sortie en ouvrant le port, tout ce que vous avez à faire est de mettre ce que vous voulez envoyer dans ce buffer, en utilisant la fonction tp__send(). Dès qu'une chaîne est dans le buffer, les fonctions internes de TELEPATHY et l'UART prennent en charge l'information et surveillent son bon acheminement en contrôlant les niveaux de flux.

Toutes les fonctions de TELEPATHY contiennent un code interne d'erreur. Quelques fonctions retournent un résultat d'erreur directement, mais toutes établissent un code d'erreur avant de quitter l'exécution.

La fonction tp__error() retourne le code d'erreur de la dernière fonction appelée. Cela vous permet de tester la bonne fin de chaque fonction de communication. La fonction tp__errmsg() retourne une description commentée du code d'erreur retourné par tp__error.

Des codes d'erreur sont générés pour les erreurs générales, les erreurs de transmission de fichier, et les erreurs du DOS. Un fichier d'en-tête est fourni, avec la définition des constantes manifestes pour ces codes d'erreur.

Un autre avantage de TELEPATHY qui peut être groupé dans la gestion des erreurs, est sa possibilité de "débrancher" ses interruptions lors d'une sortie forcée du programme. Si votre programme se plante lamentablement, vous n'avez pas à vous soucier de restaurer les interruptions dans leur état initial.

La documentation ne suit pas le format de Nantucket : syntaxe, arguments, valeur retournée, description, exemples, etc... A la place, la syntaxe est démontrée avec une variable d'assignation, qui est ensuite décortiquée en même temps que la passage des paramètres.

En voici un exemple :

tp__recv() - reçoit une chaîne

cString = tp__recv(nPort[, nLength[, nTimeout]])

La valeur renvoyée par cette fonction est documentée selon la notation Hongroise de cChaîne, et par des marques au sujet de cChaîne dans le reste de de la documentation de la fonction. Il est facile de savoir ce que vous allez récolter en regardant la variable.

L.D.Q.S., l'importateur Français, assure un service après-vente direct. Le serveur AQUARIUM, sera également un relais utile, pour ses abonnés. TELEPATHY est livré en plusieurs versions : Summer 87, Clipper 5.01.

La librairie existe en 2 versions : une version classique, qui doit résider en racine, et une version prévue pour les linkers dynamiques comme BLINKER, également distribué par L.D.Q.S.

Il faut rappeler que RTLINK, le linker actuellement livré avec Clipper 5.x, n'est pas capable de mettre en overlay les modules développés en C ou ASM.

```
/*
        Les couleurs dans TBrowse
        Dominique Bruyère

*/

#include "Box.ch"
#include "Inkey.ch"

// Exemple des réactions de Colorspec
FUNCTION main
SETBLINK(.F.)
SET DATE FRENCH
CLS
test1()
test2()

IF ! FILE("test.dbf")
   createst()
ENDIF

// Exemple des méthodes deHilite() et hilite()
BROWSER("test")
SETCOLOR("+W/*B")
CLS
// Exemple avec les variables d'instances
// defColor et colorBlock.
TBROWSER("test")
SETCOLOR("+W/*B")
CLS
// Exemple avec un freeze
FBROWSER("test")
SETCOLOR("+W/*n")
CLS
// Exemple avec ColorRec()
RBROWSER("test")
SETCOLOR("+W/*B")
CLS
RETURN nil
* eof démo couleurs dans TBrowse
*********************************************
```

Figure n° 1

```
FUNCTION test1
LOCAL b := TBrowseDB(0, 0,MAXROW(),MAXCOL())
/*
 Nous avons créé un objet TBrowse absolument vide.
 sa variable ColorSpec est chargée par le
 SETCOLOR() par défaut.
 A signaler: une petite erreur dans le Norton
 les fonctions PADx on besoin d'un SAY devant !
*/

@ 8,0 SAY padc("Test 1 "+b:colorSpec,80) COLOR b:colorSpec

/* une petite modification */
b:colorSpec := 'W/B,+W/B,+W/B,+GR/R'
@ 10,0 say PADC("Test 1 modifié "+b:colorSpec,80);
 COLOR b:colorspec
RETURN nil

FUNCTION test2
LOCAL b
/* dans la fonction test2, le SETCOLOR()
   est modifié avant de créer notre objet
*/
SETCOLOR('r*/+w, +w/b, w/r, +gr/r, +w/rb')
b := TBrowseDB(0, 0, MAXROW(), MAXCOL())
@12,0 SAY PADC("Test 2 "+b:colorSpec,80);
       COLOR b:colorspec
@24, 0 SAY PADC("Appuyez sur une touche pour continuer",80)
INKEY(0)
CLS
RETURN nil
*************************************************************
// Exemple des méthodes deHilite() et hilite()
// et test sur des enregistrements deleted(.T.)


FUNCTION browser(dbf_file)
LOCAL i, b := TBrowseDB(2,20, 20, 59)
LOCAL  c, touche := 0, unit, pseudo, debut,duree
LOCAL del1:=10, del2:=20, del3:=30
//les touches minimum pour un test

LOCAL btouche :=;
   { { K_UP,    { || b:up() } },;
     { K_DOWN, { || b:down() } },;
     { K_PGUP, { || b:Pageup() } },;
     { K_PGDN, { || b:pagedown() } },;
     { K_LEFT, { || b:left() } },;
     { K_RIGHT,{ || b:right() } } }
```

Figure n° 2

```
/* ouverture du fichier test en spécifiant
 new avec .T.,le driver utilisé, bientôt nous
 en aurons d'autres...et en utilisant l'alias.
*/

DBuseArea( .T., "DBFNTX", dbf_file,"pseudo")
DBgoto(del1); DBdelete()
DBgoto(del2); DBdelete()
DBgoto(del3); DBdelete()
pseudo->(DBgotop())
b:colorSpec := "+W/B,+GR/B,W/RB,N/GR*,W/RB,+GR/RB"+;
               "+W/B*,+B/W*"

FOR i = 1 TO FCOUNT()
   c = TBColumnNew(field(i),FIELDWBLOCK(field(i),SELECT()))
   c:colorBlock:={|x|if(deleted(),{4,3 },{1,2})}
   b:AddColumn( c )
NEXT

b:autoLite = .f.
@ 01,19,21,60 BOX B_DOUBLE COLOR "+W/B"
@ 23,0 SAY PADC("Les enregistrements effacés sont visibles",80);
           COLOR "+*GR/B"


DO WHILE touche != K_ESC
   debut := seconds()
   DO WHILE ! b:stabilize() .AND. (touche:=INKEY())==0
   ENDDO
   duree := seconds() - debut
   debut := 0
   @ 0,2 Say duree
   IF b:stable
      b:hilite()
      touche = INKEY(0)
      b:deHilite()
   ENDIF
   /* Les trois lignes de code ci-dessous
      éliminent toutes la boucle DO CASE
      ENDCASE.
   */
   IF (unit:=ASCAN(btouche,{|a|touche==a[1]}))>0
      EVAL(btouche[unit, 2])
   ENDIF
ENDDO
   RECALL ALL
   pseudo->(DBclosearea())
RETURN nil
```

Figure n° 3

```
// Exemple avec les variables d'instances
// defColor et colorBlock.

FUNCTION Tbrowser(dbf_file)

LOCAL i, b := TBrowseDB(2,20, 20, 59)
LOCAL  c, touche := 0, unit, pseudo,x

/*
  les touches minimum pour un test
*/

LOCAL btouches := { { K_UP, { || b:up() } },;
                    { K_DOWN, { || b:down() } } , ;
                    { K_PGUP, { || b:Pageup() } },;
                    { K_PGDN, { || b:pagedown() } } ,;
                    { K_LEFT, { || b:left() } },;
                    { K_RIGHT,{ || b:right() } } }
SETCOLOR("W/B")
@ 01,19,21,60 BOX B_DOUBLE COLOR "+W/B"
DBuseArea( .T., "DBFNTX", dbf_file,"pseudo")
/* mise en place du colorspec.
  Pour rassurer d'éventuels futurs clients,
  je tiens à préciser que je n'emploie jamais
  de palette de couleurs aussi laides !
*/
*                 1    2    3    4    5     6
b:colorSpec := "+W/B,+GR/B,W/RB,N/GR*,+W/R,+GR/RB"+;
               "+W/B*,+B/W*"

FOR i := 1 TO FCOUNT()
   c := TBColumnNew(FIELD(i), FIELDWBLOCK(FIELD(i), SELECT()))
   c:defColor := {7, 8}
   // les numériques négatifs sont mis en couleur burffff
   IF TYPE(field(i)) == "N"
      c:colorBlock := { | x | if(x < 0, { 4, 3 } , { 1, 2} ) }
   ENDIF
   // la date système est mise en couleur smurglb
   IF TYPE(field(i)) == "D"
      c:colorBlock:={|x|if(x==date(),{5,6},{1,2})}
   ENDIF
   b:addcolumn(c)
NEXT

@ 22,0 SAY PADC("La date système apparait dans"+;
  " une couleur spéciale",80)  COLOR "+*GR/B"
@ 23,0 SAY PADC("Les sommes négatives sont"+;
 " repérées par une autre couleur",80) COLOR "+*GR/B"
 *              Figure n° 4
```

```
DO WHILE touche != K_ESC
   DISPBEGIN() // écriture dans l'écran virtuel
   DO WHILE ! b:stabilize() .AND. (touche:=INKEY())==0
   ENDDO
   DISPEND()

   IF b:stable
      touche = INKEY(0)
   ENDIF
   IF (unit:=ASCAN(btouches,{|a|touche==a[1]}))>0
      EVAL(btouches[unit, 2])
   ENDIF
ENDDO
// fermeture du fichier; notez l'utilisation de l'alias.

pseudo->(DBClosearea())
RETURN nil
*eof defcolor et colorblock
************************************************************
// Exemple d'un freeze avec changement de couleur


FUNCTION FBrowser(dbf_file)

LOCAL i, b := TBrowseDB(2,20, 20, 59)
LOCAL  c, touche := 0, unit, pseudo,x, debut,duree
LOCAL colfreeze := 1  // première colonne

LOCAL btouches := { { K_UP, { || b:up() } },;
                    { K_DOWN, { || b:down() } } , ;
                    { K_PGUP, { || b:Pageup() } },;
                    { K_PGDN, { || b:pagedown() } } ,;
                    { K_LEFT, { || b:left() } },;
                    { K_RIGHT,{ || b:right() } } }
SETCOLOR("W/B")
SET DATE FRENCH
DBuseArea( .T., "DBFNTX", dbf_file,"pseudo")
@ 01,19,21,60 BOX B_DOUBLE COLOR "+W/B"
b:colorSpec := "+W/B,+GR/B,W/RB,N/GR*,W/RB,+GR/RB"+;
               "+W/B*,+B/W*"

FOR x := 1 TO FCOUNT()
   c:=TBColumnNew(field(x),fieldwblock(field(x),SELECT()))
   b:freeze := colfreeze
   c:defcolor :=IF( x == colfreeze,{4, 5},{5,6})
   b:AddColumn( c )
NEXT
    *
```

Figure n° 5

```
   @ 22,0 SAY PADC("Couleur différente pour la"+;
      " colonne 'freezée'",80)
   DO WHILE touche != K_ESC
      /* blocage du curseur en dehors du freeze */
      IF b:colPos <= b:freeze
         b:colPos := b:freeze + 1
      ENDIF
      debut := seconds()
      DO WHILE ! b:stabilize() .AND. (touche:=inkey())==0
      ENDDO
      duree := seconds() - debut
      debut := 0
      @ 0,2 Say duree

      IF b:stable
         TOUCHE := inkey(0)
      ENDIF
      IF (unit:=ASCAN(btouches,{|a|touche==a[1]}))>0
         EVAL(btouches[unit, 2])
      ENDIF
   ENDDO
   pseudo->(DBClosearea())
   RETURN nil
   *eof freeze
   ************************************************************

   /* Exemple d'utilisation de colorRect().
      Le but de la manoeuvre est, un peu comme
      dans un traitement de texte, de marquer un
      bloc sur l'écran, pour le déplacer. Ici, le
      déplacement se fera en direction d'un autre
      fichier, créé à la commande, avec les seuls
      les champs marqués et les enregistrements
      compris entre les limites d'affichage
      supérieure et inférieure. Dommage !
      Par contre les déplacements latéraux sont
      pris en compte et entrent dans colorRect()
      Une application possible de cette fonction
      est la création d'un fichier temporaire
      pour un mail merge dont le critère de
      tri ne soit pas cohérent. Le travail se
      fait sur le fichier temporaire et donc
      libère le fichier maître.
      Il suffit une fois que le fichier est créé
      de faire des DBappend().
      Blocked indique si nous sommes en train de
      construire le bloc à exporter.    */
*                  Figure  n° 6
```

```
#define C_NON 0   // curseur niet
#define C_OUI 1   // curseur ok
#define C_PLEIN 3 // big curseur

FUNCTION RBrowser(dbf_file)
// notre objet
LOCAL browse := TBrowseDB(0, 0, maxrow() - 1, maxcol())
LOCAL x, c, touche := 0, blocked := .f., unit
// coordonnées du bloc à marquer
LOCAL h:=0,g:=0,b:=0,d:=0,newstruc,oldstruc

DBuseArea( .T., "DBFNTX", dbf_file,"pseudo")
CLS

browse:ColorSpec := '+gr/n, +w/*r, n/*gr'
*browse:colorSpec = 'w/n, w/r, n/w'
FOR x = 1 TO FCOUNT()
   c = TBColumnNew(FIELD(x), fieldwblock(FIELD(x), SELECT()))
   c:defColor = {1, 3}
   browse:AddColumn( c )
NEXT
SETCURSOR(C_NON)
@ MAXROW(),0 SAY PADC("[Entrée] pour marquer"+;
   " le début et la fin du bloc",80)
DO WHILE touche != K_ESC
   DO WHILE ! browse:stabilize() .AND. (touche:=INKEY())==0
   ENDDO
   IF browse:stable
      touche = inkey(0)
   ENDIF
   DO CASE
      CASE touche == K_UP
         browse:up()
         /*
          si l'opération a commencé et que la surface
          a diminué, on repeint le bloc en couleur
          bloquifiante et on modifie ses coordonnées.
         */
         IF blocked
            IF browse:rowPos <= b .AND. b > h
               browse:colorRect( { h, g, b--, d }, {1, 1} )
            ENDIF
         ENDIF
      CASE touche == K_DOWN
         // comme plus haut, à l'envers...
         browse:down()
         IF blocked .and. browse:rowPos = b
            b++
         ENDIF
*              Figure n° 7
```

```
      CASE touche == K_LEFT
         browse:left()
         /*
          si l'opération a commencé et que la surface
          a diminué, on repeint le bloc en couleur
          bloquifiante.
         */
         IF blocked
            IF browse:colPos < d .and. d > g
               browse:colorRect( { h, g, b, d-- }, {1, 1} )
            elseif g > 1
               g--
            ENDIF
         ENDIF

      CASE touche == K_RIGHT
         browse:right()
         /*
          si l'opération a commencé et qu'il reste des
          colonnes à bloquer....
         */
         IF blocked .and. browse:colPos <= browse:colCount
            d++     // augmentation de la zone bloquifiée...
         ENDIF

      CASE touche == K_ENTER
         IF ! blocked     // l'opération commence
            h := b := browse:rowPos    // où suis-je
            g := d := browse:colPos    // et dans quel état ...
            blocked = .T. // l'opération a commencé
            setcursor(C_PLEIN)  // BIG curseur
         ELSE
            /*
               On est en plein suspense et on voudrait bien
               voir si ce machin fonctionne. Le rôle de
               newstruc et de oldstruc sera compris, même
               par ceux qui ont fait bélouchistanais en
               première langue.
            */
            newstruc := {}           // vide
            oldstruc := dbstruct()   // plein
   /*
      On va de gauche à droite pour n'oublier personne.
      On met dans le tableau vide, la structure des champs
      concernés et on passe le tout à DBcreate.
      Enfin on ouvre la petite nouvelle.
      Je peux vous le refaire en Summer 87, si vous avez
      du temps libre !
                     Figure n° 8                          */
```

```
            FOR x = g TO d
               AADD(newstruc, oldstruc[ ASCAN(oldstruc, ;
                {|unit| browse:GETCOLUMN(x):heading==unit[1]})])
            NEXT

            DBcreate("tempo", newstruc)

            DBuseArea( .T., "DBFNTX", "tempo","tpseudo")

            /*
               On remet les pendules à l'heure et le pointeur
               en haut de la zone qui nous intéresse.
               N'oubliez pas de réciter 2 code-blocks et
               3 TBrowse avant de vous endormir. Tous vos
               bugs vous seront pardonnés.
            */
            pseudo->(GOskip(h - browse:rowPos))

            /* Le début de la boucle infernale */

      FOR x = h TO b
         tpseudo->(DBappend()) // Voir NG
         /* repèrage des colonnes bloquifiées ???
            Quand je pense à la gymnastique qu'il fallait
            faire avec 87 pour un résultat pareil !
         */
         FOR c = g TO d
            FIELDPUT(c - g + 1, EVAL(browse:getColumn(c):block))
         NEXT
         pseudo->(GOskip(1))
      NEXT
      CLS
      DBgotop()
      dbedit(0,0,24,79)
      tpseudo->(DBClosearea())
      touche = K_ESC
     ENDIF
   ENDCASE
   IF blocked // merci M. Ripolin
      browse:ColorRect( { h, g, b, d }, {2, 2} )
   ENDIF
ENDDO
Setcursor(C_OUI)
pseudo->(DBClosearea())
RETURN nil
```

Figure n° 9

```
FUNCTION GOskip(rgts)
DBskip(rgts)
RETURN nil
// Mise en place du fichier pour les test
// avec des valeurs variées et des prénoms
// bas-bélouchistanais
    FUNCTION createst
    FIELD nom, date, cotisation, nouveau, notes IN test
    LOCAL i, atest:={},message:="Création du fichier test"
    SETCOLOR("W/B,W/R,+W/B*")
    CLS
 //      ajout dans le tableau des caractéristiques
//       des champs du fichier TEST.DBF.
    AADD(atest,{"NOM", "C", 10, 0})
    AADD(atest,{"DATE", "D", 8, 0} )
    AADD(atest,{"COTISATION", "N", 10, 2})
    AADD(atest,{"NOUVEAU","L",1,0} )
    AADD(atest,{"NOTES","M",10,0})
    // création du fichier .DBF
    DBcreate("test", atest)
    DBuseArea( .T., "DBFNTX", "TEST")

    @ 9,0 say PADC(message,80)
    @ 11,10,13,63 BOX B_DOUBLE COLOR "W/R"
    FOR i := 1 TO 26
      test-> (DBAppend())
      test->nom  := "JEAN"+CHR(64+i)
      test->date := DATE() - 10+i
      test->cotisation := ;
          IF( DAY(date)%2 == 0,-10000+(i*3),10000+(i*7))
      test->nouveau :=;
       IF( RECNO() >=10 .AND. RECNO()<=22,.T.,.F.)
      test->notes := "Vide"
    @ 12,10+i say" " COLOR "W/R"
    NEXT
    FOR i := 27 TO 52
      test-> (DBAppend())
      test->nom  := "JEANZ"+CHR(64+i-26)
      test->date := DATE() - 10+i
      test->cotisation := ;
      IF( DAY(date)%2 == 0,-10000+(i*3),10000+(i*7))
      test->nouveau := .F.
      test->notes := "Vide"
    @ 12,10+i say" " COLOR "W/R"
    NEXT
```

Figure n° 10

```
    /* création du fichier .NTX */
    dbCreateInd( "test", "nom", {|| nom},NIL )
    @ 09,0 say PADC("                                ",80)
    @ 11,10 CLEAR TO 13,63
    test->(DBClosearea())
    CLS
    RETURN nil
```

Figure n° 11

Le Multimédia et nouveaux types de communication

XI*eme*
Congrès de l'E.A.O. et du Multimédia

Les 7-8-9 Janvier 1992
Palais des Congrès
Porte Maillot (Paris 17e)

1
Exposition
2
Colloques animés par des intervenants de haut niveau sur 3 jours

Premier colloque :

Quels Systèmes de Formation avec l'Enseignement Assisté par Ordinateur (E.A.O.) ? Quelles Stratégies d'insertion et coûts des Nouvelles Technologies de l'Education ? Conception, Outils pour la création de Multimédia Interactif

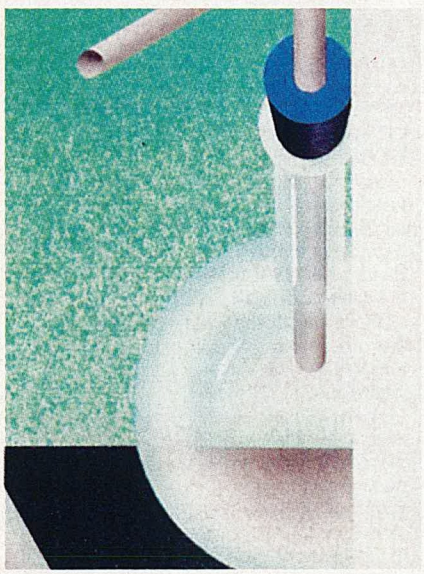
Deuxième colloque :

Guerre des Standards, Quelles Normes pour les années 90 ? Quelles Applications du Multimédia et de la Communication Interactive ? et Quels choix des Outils pour la création de Multimédia Interactif ?

Renseignements et inscriptions
Le Journal de la Formation Continue et de l'E.A.O.
2 rue d'Amsterdam
75009 Paris
Tél. : (1) 47 64 07 57 - (1) 42 67 93 80

VGA s'imposant comme le ciment de toutes les applications graphiques, nous allons lui construire une librairie en béton

Jean-Michel SPINER



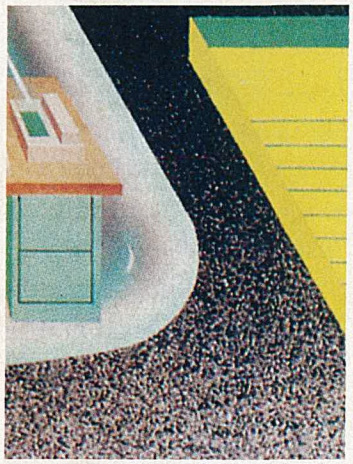
Encore récemment réservée aux machines haut de gamme, la norme VGA apparaît bel et bien comme un standard appelé à durer.

De nombreux constructeurs en équipent directement leurs machines sur la carte mère, et l'étape suivante, celle des cartes graphiques intelligentes, dotées d'un processeur et de fonction câblée, se verra certainement imposée la compatibilité descendante VGA.

Le soft ayant toujours eu un temps de retard sur le matériel, l'utilisateur de base travaille aujourd'hui sur une machine qui a de réelles possibilités graphiques, mais peu de logiciels aptes à les exploiter.

Programmeurs, à vos claviers, nous allons commencer à rédiger une librairie de fonctions graphiques dédiées au mode VGA, qui sera séparée en trois groupes : les fonctions élémentaires, les fonctions un peu plus évoluées (tracé de cercle, de vecteurs,...), et les autres, de haut niveau (fenêtrage, animation,...).

Lorsque l'on sait que la plupart des logiciels passent 80 % de leur temps d'activité (hos attente clavier) à gérer l'affichage, on comprend qu'il faille attacher un soin tout particulier à la rapidité des fonctions.



Nous avons constaté que les différences de performances se font essentiellement sur le mode d'adressage de la mémoire vidéo (BIOS ou accès direct), et non sur le langage de programmation employé.

C'est pourquoi certaines fonctions qui n'auraient que peu gagnées à être en assembleur figureront en C, afin d'assurer une meilleure lecture.

Pour s'en convaincre, examinons les résultats d'un test très simple qui consiste à remplir 3 fois l'écran en mode 13h en bouclant sur tous les points.

```
BIOS / C     : 51s
BIOS / ASM   : 46s
DIRECT / C   : 12s
DIRECT / ASM :  9s
```

Le court-circuitage du BIOS ne se fera que dans les cas où il est rentable.

Prenons l'exemple de Set_Screen_mode, fonction permettant l'initialisation de l'écran.

Dans la mesure où cette fonction n'est appelée qu'une fois par application, il semble bien inutile de tenter de grignoter quelques millisecondes en se compliquant inutilement l'existence.

Il nous reste encore à évacuer une vieille crainte : ne risque-t-on pas, en utilisant l'adressage direct, de mettre en péril la compatibilité et la pérennité de nos applications ?

S'il est incontestable que le BIOS offre certaines garanties à ce sujet, on s'imagine mal un constructeur mettre sur le marché une carte graphique qui ne se conformerait pas aux adressages actuellement en vigueur.

En effet, il interdirait ainsi le fonctionnement des principaux logiciels, ce qui limiterait le produit à une diffusion confidentielle.

La rapidité d'exécution entraîne la contrainte de spécialisation du code.

Il est très pénalisant, à chaque affichage de pixel, de se demander qui on est, ce qu'on nous veut, quel est le mode graphique, la taille de l'écran, l'adresse des registres, et la marque du lecteur de disquette.

La suppression de ce genre de test conduit à une inévitable prolifération de fonctions.

En fait, pour une même action, il y aura en général autant de fonctions différentes que de modes graphiques à assurer, plus une pour le BIOS.

C'est cette classification que nous allons garder pour séparer les modules.

vgakern.c : contient les variables d'affichage, la fonction d'initialisation, et, d'une façon générale, les fonctions qui ne dépendent pas du mode d'affichage.

vgabios.c : fonctions d'affichages utilisant le BIOS, de Put_pixel_bios à Circle_bios.

vgac12h.c et vgaa12h.asm : fonctions d'affichages par accès direct en mode 12h (640 × 480 en 16 couleurs).

vgac13h.c et vgaa13h.asm : fonctions d'affichages par accès direct en mode 13h (320 × 200 en 256 couleurs).

VGAKERN : le noyau de la librairie

Ce module, assez mince pour l'instant, sera appelé à s'étoffer au fur et à mesure de nos progrès dans les fonctions évoluées.

Il devra être systématiquement linké avec l'application si celle-ci utilise un autre module de la librairie.

On y trouve la variable ptr_video_memory, pointeur sur l'adresse de la mémoire vidéo et définit de cette façon : unsigned char huge *ptr_video_memory.

Le mot réservé huge, en MSC, désigne un pointeur sur une zone qui n'est pas dans le segment data ordinaire, et dont la taille peut être supérieure à 64 Ko.

Les fonctions de vgakern sont les suivantes :

```
short Get_graph_mode ();
short Init_graph (short graph_mode);
void Set_graph_mode (short graph_mode);
void Set_overscan_color (short overscan_color);
```

Set_graph_mode utilise la fonction 0 de l'interruption bios video (10h) pour mettre l'affichage dans le mode désiré. Les modes possibles sont les suivants :

mode	résolution	couleurs
0,1	texte 40x25	16
2,3	texte 80x25	16
4,5	320x200	4
6	640x200	2
7	texte 80x25	2
D	320x200	16
E	640x200	16
F	640x350	2
10	640x350	16
11	640x480	2
12	640x480	16
13	320x200	256

Init_graph initialise le segment de ptr_video_memory en fonction du mode graphique utilisé.

Le mode choisi est ensuite transmis à Set_graph_mode et Get_graph_mode, se charge de vérifier si le mode a été accepté par la carte. Il retourne une valeur négative dans le cas inverse.

Enfin, Set_overscan_color est employé pour choisir la couleur de bordure, c'est-à-dire la surface comprise dans les limites physiques de l'écran, mais non adressables.

VGABIOS : le module BIOS

Chargé des affichages élémentaires avec accès standard par le BIOS, ce module contient les fonctions suivantes :

```
void Circle_bios
(short centx,short centy,short rayon,short coul);
void Circle_f_bios
(short centx,short centy,short rayon,short coul);
void Line_bios
(short x1,short y1,short x2,short y2,short coul);
void Put_pixel_bios
(short x,short y,short couleur);
void Rect_bios
(short x1,short y1,short x2,short y2,short coul);
void Rect_f_bios
(short x1,short y1,short x2,short y2,short coul);
```

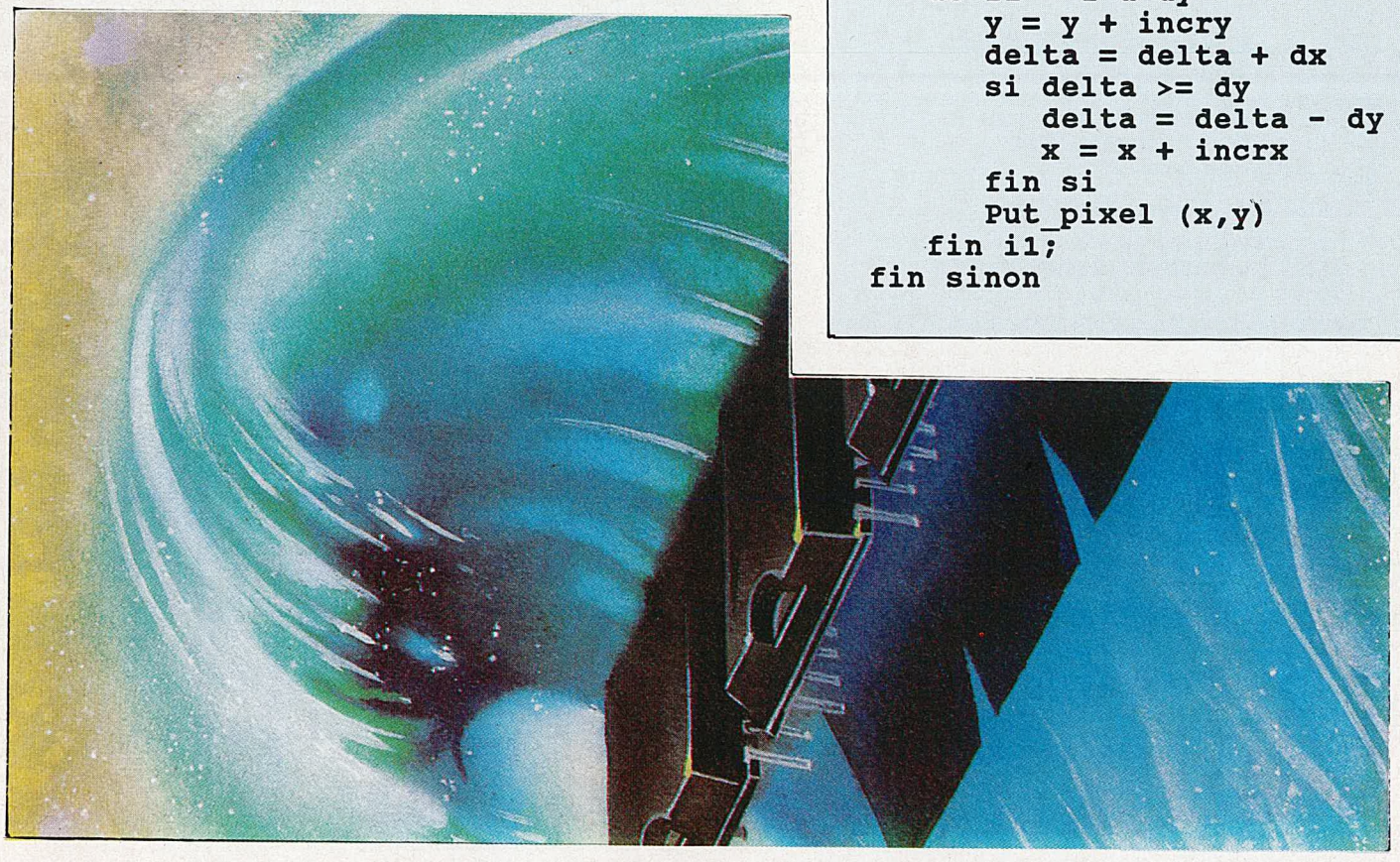
Circle_bios et Circle_f_bios sont respectivement utilisées pour le tracé de cercles et de cercles pleins.

Algorithme de tracé de cercle.

```
x = rayon
y = 0
c = 3 - 2*rayon
tant que x>=y
   si x!=y
      Put_Pixel (centx+x,centy+y)
      Put_Pixel (centx+y,centy-x)
      Put_Pixel (centx-x,centy-y)
      Put_Pixel (centx-y,centy+x)
   fin si
   si y!=0
      Put_Pixel (centx+x,centy-y)
      Put_Pixel (centx-y,centy-x)
      Put_Pixel (centx-x,centy+y)
      Put_Pixel (centx+y,centy+x)
   fin si
   si c<0
      c = c + 4*y + 6
   fin si
   sinon
      c = c - 4*x + 4*y + 10
      x = x - 1
   fin sinon
   y = y + 1
fin tant que
```

Line_bios, tracé de vecteur, utilise l'algorithme classique de Bresenham.

```
incrx = 1
incry = 1
x = x1
y = y1
Put_pixel (x,y)
si x1 > x2
   incrx = -1
fin si
si y1 > y2
   incry = -1
fin si
dx = abs(x1-x2)
dy = abs(y1-y2)
si dx > dy
   delta = dx / 2
   de i1 = 1 à dx
      x = x + incrx
      delta = delta + dy
      si delta >= dx
         delta = delta - dx
         y = y + incry
      fin si
      Put_pixel (x,y)
   fin i1
fin si
sinon
   delta = dy / 2
   de i1 = 1 à dy
      y = y + incry
      delta = delta + dx
      si delta >= dy
         delta = delta - dy
         x = x + incrx
      fin si
      Put_pixel (x,y)
   fin i1;
fin sinon
```



Ce tracé de droite inclut les points extrêmes. Le passage par le BIOS rend inutile toute tentative d'optimisation sur les droites horizontales ou verticales.

Enfin, Rect_bios et Rect_f_bios traceront les rectangles.

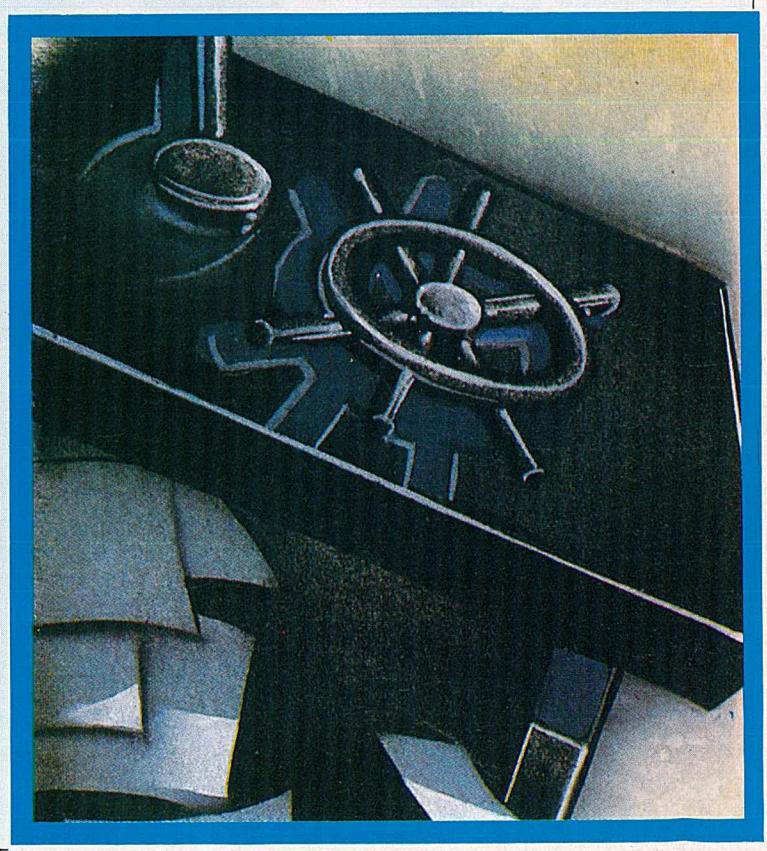
De la même façon que pour le tracé de vecteur, l'usage du BIOS ne permet l'accès à la mémoire vidéo que pour le traitement d'un pixel unique.

Il n'y a donc pas de solutions plus rapides que la double boucle pour traiter Rect_f.

Nous verrons le mois prochain des moyens beaucoup moins standards, mais beaucoup plus efficaces pour tracer des figures élémentaires dans les modes 12h et 13h.

```
10/09/1991  02:31        FICHIER  :  VGAKERN.C

/*
      +-----------------------------------------+--------+
      !                  VGAKERN.C              !  1.00  !
      +-----------------------------------------+--------+
      !                                                   !
      !      ± FONCTION    :   Noyau lib VGA              !
      !                                                   !
      !      ± AUTEUR      :   Jean-Michel SPINER         !
      !                                                   !
      +---------------------------------------------------+
*/


#include <dos.h>


short Get_graph_mode ();
short Init_graph (short graph_mode);
void Set_graph_mode (short graph_mode);
void Set_overscan_color (short overscan_color);


#define INT_VIDEO 0x10
#define VGA1 0x12  /* VGA Mode 12H (640x480x16)   */
#define VGA2 0x13  /* VGA Mode 13H (320x200x256)  */


unsigned char huge *ptr_video_memory;


/*                GET_GRAPH_MODE                    */
short Get_graph_mode ()
{
   union REGS inregs,outregs;

   inregs.h.ah = 0x0F;
   int86 (INT_VIDEO,&inregs,&outregs);
   return (outregs.h.al);
}


/*                INIT_GRAPH                        */
short Init_graph (graph_mode)
short graph_mode;
{

   switch (graph_mode) {
      case VGA1 :
      case VGA2 :
         FP_SEG (ptr_video_memory) = 0xA000;
         break;
      default :
         return (-1);
   }
   Set_graph_mode (graph_mode);
```

```
10/09/1991  02:31        FICHIER  :  VGAKERN.C

   if (Get_graph_mode() != graph_mode) return (-2);
   return (0);
}


/*                SET_GRAPH_MODE                    */
void Set_graph_mode (graph_mode)
short graph_mode;
{
   union REGS inregs,outregs;

   inregs.h.ah = 0;
   inregs.h.al = (unsigned char)graph_mode;
   int86 (INT_VIDEO,&inregs,&outregs);
}


/*                SET_OVERSCAN_COLOR                */
void Set_overscan_color (overscan_color)
short overscan_color;
{
   union REGS inregs,outregs;

   inregs.h.bh = overscan_color;
   inregs.h.ah = 0x10;
   inregs.h.al = 0x01;
   int86 (INT_VIDEO,&inregs,&outregs);
}
```



```
/*
      +--------------------+--------------------+--------+
      !                  VGABIOS.C               !  1.00  !
      +-----------------------------------------+--------+
      !                                                    !
      !      ± FONCTION   :   lib VGA                      !
      !                       fonctions de trac BIOS       !
      !                                                    !
      !      ± AUTEUR     :   Jean-Michel SPINER           !
      !                                                    !
      +----------------------------------------------------+
*/



#include <dos.h>



void Circle_bios (short centx,short centy,short rayon,short coul);
void Circle_f_bios (short centx,short centy,short rayon,short coul);
void Line_bios (short x1,short y1,short x2,short y2,short coul);
void Put_pixel_bios (short x,short y,short couleur);
void Rect_bios (short x1,short y1,short x2,short y2,short coul);
void Rect_f_bios (short x1,short y1,short x2,short y2,short coul);



/*                CIRCLE_BIOS                    */
void Circle_bios (centx,centy,rayon,coul)
short centx,centy,rayon,coul;
{
   register short i1;
   short x,y,c;

   x = rayon;
   y = 0;
   c = 3 - 2*rayon;
   while (x>=y) {
      if (x!=y) {
         Put_Pixel_bios (centx+x,centy+y,coul);
         Put_Pixel_bios (centx+y,centy-x,coul);
         Put_Pixel_bios (centx-x,centy-y,coul);
         Put_Pixel_bios (centx-y,centy+x,coul);
      }
      if (y!=0) {
         Put_Pixel_bios (centx+x,centy-y,coul);
         Put_Pixel_bios (centx-y,centy-x,coul);
         Put_Pixel_bios (centx-x,centy+y,coul);
         Put_Pixel_bios (centx+y,centy+x,coul);
      }
      if (c<0) {
         c = c + 4*y+6;
      }
      else {
         c = c - 4*x + 4*y + 10;
         x--;
      }
      y++;
   }
}
```



```
/*                CIRCLE_F_BIOS                  */
void Circle_f_bios (centx,centy,rayon,coul)
short centx,centy,rayon,coul;
{
   register short i1;
   short x,y,c;

   x = rayon;
   y = 0;
   c = 3 - 2*rayon;
   while (x>=y) {
      if (x==y && x!=0 && y!=0) {
         Line_bios (centx-x,centy+y,centx+x,centy+y,coul);
         Line_bios (centx-x,centy-y,centx+x,centy-y,coul);
      }
      if (x!=y && x!=0 && y==0) {
         Line_bios (centx-x,centy+y,centx+x,centy+y,coul);
         Line_bios (centx-y,centy-x,centx+y,centy-x,coul);
         Line_bios (centx-y,centy+x,centx+y,centy+x,coul);
      }
      if (x!=y && x!=0 && y!=0) {
         Line_bios (centx-x,centy+y,centx+x,centy+y,coul);
         Line_bios (centx-y,centy-x,centx+y,centy-x,coul);
         Line_bios (centx-x,centy-y,centx+x,centy-y,coul);
         Line_bios (centx-y,centy+x,centx+y,centy+x,coul);
      }
      if (c<0) {
         c = c + 4*y+6;
      }
      else {
         c = c - 4*x + 4*y + 10;
         x--;
      }
      y++;
   }
}



/*                LINE_BIOS                      */
void Line_bios (x1,y1,x2,y2,coul)
short x1,y1,x2,y2,coul;
{
   register short i1;
   short incrx,incry,delta;
   short dx,dy;
   short x,y;

   incrx=1;
   incry=1;
   x=x1;
   y=y1;
   Put_pixel_bios (x,y,coul);
   if (x1 > x2) incrx=-1;
   if (y1 > y2) incry=-1;
   dx=abs(x1-x2);
   dy=abs(y1-y2);
   if (dx > dy) {
      delta=dx / 2;
      for (i1=1;i1<=dx;i1++) {
         x += incrx;
         delta += dy;
         if (delta >= dx) {
```

10/09/1991 02:30 FICHIER : VGABIOS.C

```
            delta -= dx ;
            y += incry;
         }
         Put_pixel_bios (x,y,coul);
      }
   }
   else {
      delta=dy / 2;
      for (i1=1;i1<=dy;i1++) {
         y += incry;
         delta += dx;
         if (delta >= dy) {
            delta -= dy;
            x += incrx;
         }
         Put_pixel_bios (x,y,coul);
      }
   }
}



/*                PUT_PIXEL_BIOS                */
void Put_pixel_bios (x,y,couleur)
short x,y,couleur;
{
   union REGS inregs,outregs;

   inregs.h.ah = 0x0C;
   inregs.h.al = (unsigned char)couleur;
   inregs.h.bh = 0x00;
   inregs.x.cx = x;
   inregs.x.dx = y;
   int86 (0x10,&inregs,&outregs);
}
```

```
/*                RECT_BIOS                     */
void Rect_bios (x1,y1,x2,y2,coul)
short x1,y1,x2,y2,coul;
{
   Line_bios (x1,y1,x1,y2,coul);
   Line_bios (x1,y2,x2,y2,coul);
   Line_bios (x2,y2,x2,y1,coul);
   Line_bios (x2,y1,x1,y1,coul);
}



/*                RECT_F_BIOS                   */
void Rect_f_bios (x1,y1,x2,y2,coul)
short x1,y1,x2,y2,coul;
{
   register short i1;
   register short i2;
   short tmp;

   if (x1>x2) {tmp=x1; x1=x2; x2=tmp;}
   if (y1>y2) {tmp=y1; y1=y2; y2=tmp;}
   for (i1=x1;i1<=x2;i1++) {
      for (i2=y1;i2<=y2;i2++) {
         Put_pixel_bios (i1,i2,coul);


      }
   }
}
```


BON DE COMMANDE

à retourner
à la Revue de l'Utilisateur PC
2, rue d'Amsterdam
75009 Paris

Nom : .....
Société : .....
Adresse : .....
Code postal : .....
Ville : .....
Tél. : .....
Montant : 21 F × =
24 F × =
(à partir du n° 45)
Total achat revues =
+ Frais Poste =
Montant du chèque =

Règlement par chèque bancaire :
à l'ordre de SOFTin Communications : 2, rue d'Amsterdam, 75009 Paris
Frais poste pour 1 n° 7,50 F - 2 n° 14,00 F - 3/4 n° 18,00 F - 5/6 n° 25,00 F
(au-dessus par multiple)

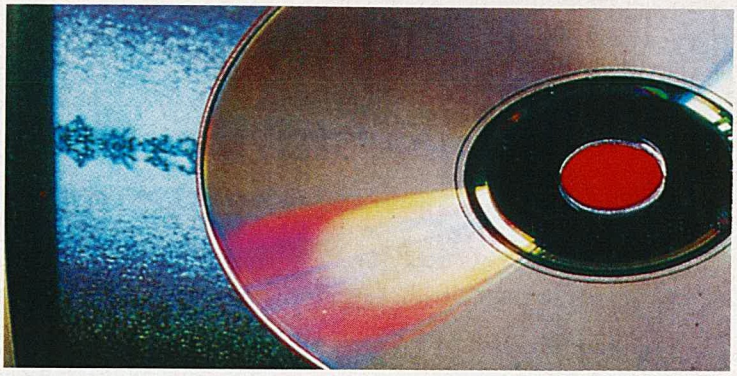
POUR COMMANDER LES ANCIENS NUMEROS

Cochez les chiffres qui vous intéressent :

☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
☐ 6	☐ 7	☐ 8	☐ 9	☐ 10
☐ 11	☐ 12	☐ 13	☐ 14	☐ 15
☐ 16	☐ 17	☐ 18	☐ 19	☐ 20
☐ 21	☐ 22	☐ 23	☐ 24	☐ 25
☐ 26	☐ 27	☐ 28	☐ 29	☐ 30
☐ 31	☐ 32	☐ 33	☐ 34	☐ 35
☐ 36	☐ 37	☐ 38	☐ 39	☐ 40
☐ 41	☐ 42	☐ 43	☐ 44	☐ 45
☐ 46	☐ 47	☐ 48	☐ 49	☐ 50
☐ 51	☐ 52	☐ 53	☐ 54	☐ 55
☐ 56	☐ 57	☐ 58	☐ 59	☐ 60
☐ 61	☐ 62	☐ 63	☐ 64	☐ 65
☐ 66	☐ 67	☐ 68	☐ 69	☐ 70
☐ 71	☐ 72			

N° 9 - 17 - 18 - 19 - 20 - 22 - 23 - 34 - 35 - 36 - 37 épuisés

Calculatrice CALC.C



Laurent LEFOUET, ESGI

Ce programme permet d'effectuer de longs calculs le plus simplement du monde. En effet, une page entière de calculs ne l'effraie pas.

Toutes les fonctions mathématiques sont possibles, des logarithmes en passant par la trigonométrie, il est en outre très facile de rajouter des fonctions dans le programme. Il permet également d'obtenir le tableau de valeur d'une équation à une inconnue x.

Fonctionnement du programme

L'utilisateur saisit le calcul ou l'équation qu'il souhaite résoudre. Cette saisie est enregistrée dans le tableau t[1000].

Après la validation, ce tableau va faire l'objet d'un décorticage minutieux afin d'obtenir des éléments compréhensibles par la fonction de calcul.

t[1000] est un tableau de caractères qui contient par conséquent les codes ascii des éléments saisis par l'utilisateur. Il faut maintenant interpréter ces codes ascii afin de dissocier les signes et fonctions des variables. Deux tableaux sont prévus à cet effet, sig[1000] et expr[1000], ils contiendront respectivement les signes et les variables.

Pour mieux comprendre ce qui se passe, nous allons prendre un exemple qui recouvre l'ensemble des cas possibles (hormis les résolutions d'équations que nous verrons plus loin).

L'utilisateur saisit donc par exemple : 4+12(−2.1+sin(1)^2/4). t[i] contient les codes ascii de cette équation, soit : 52,43,49,50,40,45,50,46,49,43,115,105, 110,40,49,41,94,50,47,52,41

4 + 1 2 (− 2.1 + sin (1)^ 2/4)

L'algorithme lit t[1000] de la façon suivante :

α=0, x=0, y=0

Il lit :

t[α]=4 => x et α sont incrémentés de 1

t[α]='+' => le signe valide le nombre qui précédait.

Par conséquent :

expr[y]=expr[y]+((t[α−(x)]−48*pow (10,(x−1))) ;

t[α−(x)]=t[α−1]= (code ascii du 4 lu avant).

En clair :

expr[0]= 0+((52−48)*10^0 = 0+4*1 = 4))

Le nombre précédent le signe α bien été retrouvé.

Vous me demanderez pourquoi un algorithme si compliqué alors qu'il suffisait de faire 52−48 pour avoir 4 ?

Je dis d'accord, mais dans ce cas où le nombre est simple, si le nombre α plusieurs chiffres, ''ça se corse''. Et comme un algorithme se doit de traiter tous les cas, on en arrive à ce genre de paradoxe, pourquoi faire simple quand on peut faire compliqué ?

Après le passage dans la boucle on obtient les choses suivantes. On α :

expr[0]=4, sig[0]=0 et expr[1]=0, sig[1]='+' x=−1, y=2

La procédure continue ensuite comme au départ.

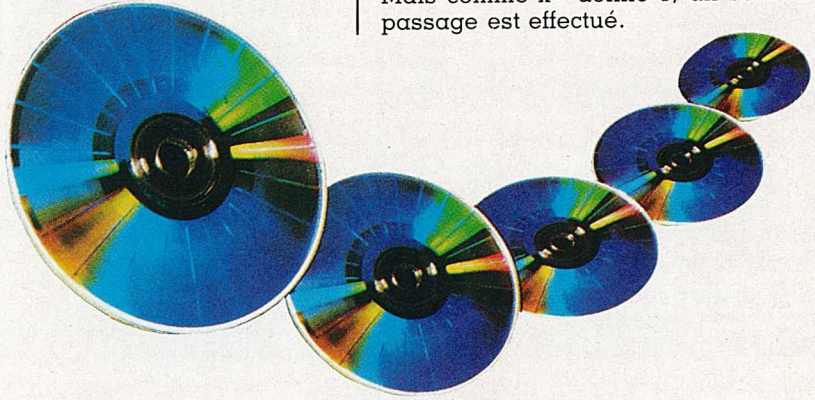
t[2]='1' x++ =0 ;
t[3]='2' x++ =1 ;
t[4]='3' x++ =2.

La parenthèse comme le signe '+' valide le nombre la précédant.

expr[2]=expr[2]+((t[α−2]−48)*10^(2−1))

expr[2]=0+(49−48)*10 = 1*10 = 10

Mais comme x-- donne 1, un second passage est effectué.



expr[2] = expr[2] + ((t[α − 1] − 48)*10^(1 − 1)

expr[2] = 10 + (50 − 48)*1 = 10 + 2*1 = 12

Une fois encore, le nombre 12 a bien été reconstitué, mais cette fois, la relative complexité de l'algorithme est justifiée.

Les mises à jour du tableau des signes sont effectuées.

Dans ce cas, il s'agit d'une factorisation, par conséquent la boucle introduit un '*' entre le nombre 12 et la parenthèse.

sig[3] = '*', sig[4] = '('.

On a maintenant les tableaux suivants :

	sig	expr
0		4
1	+	0
2		12
3	*	0
4	(	0

A la fin du processus de dissociation on obtient les tableaux finaux suivants :

sig	expr
	4
+	0
	12
*	0
(	0
	0
−	0
	2.1
+	0
s	0
i	0
n	0
(	0
	1
)	0
^	0
	2
/	0
	4
)	0
E	0

Un '0' est introduit dans le cas où un signe et aucune variable précède le nombre : (−2, (+2 ou −2)).

Le traitement du point décimal est effectué grâce à une boucle semblable à celle de reconstitution de la partie entière que nous avons vu plus haut.

Seul le calcul de expr change au niveau de la puissance de 10 (pour obtenir une partie décimale).

Les parenthèses autour du 1 ne sont pas obligatoires, on peut très bien écrire sin1.

Un 'E' marque la fin de l'expression.

On peut par conséquent noter que les signes ont bien été dissociés des variables. Chacun est à sa place.

Les emplacements libres dans le tableau expr serviront à placer les résultats intermédiaires lors des calculs. Mais nous verrons cela plus tard.

Une fois l'expression interprétée, il convient maintenant de détecter le maximum d'erreur. On commence par compter le nombre de parenthèses ouvrantes et le nombre de fermantes, s'il y a une différence, il y a forcément une erreur notée 8 ou 9.

En outre, le compte des parenthèses servira plus tard pour le respect des priorités mathématiques liées aux parenthèses.

Ensuite on recherche les erreurs du type (*,(/,+),−),*) etc..., les signes consécutifs (**,*+,/−/), les dévisions par zéro (si elles ne sont pas implicites, c'est-à-dire que l'utilisateur a réellement saisi x/0) et enfin la syntaxe des points décimaux, il n'est pas concevable qu'un nombre ait deux points décimaux.

Si l'expression est correcte, alors la machine vous confirme l'expression en affichant ce qu'elle a compris, puis le programme cherche s'il s'agit d'une équation à résoudre ou d'un simple calcul. En ce qui concerne notre exemple, il n'a pas d'erreurs et il s'agit d'un simple calcul.

Vous pourrez, si vous le souhaitez, avoir les calculs intermédiaires.

Une fois toutes ces manipulations réussies, le programme passe à la phase de calcul proprement dite.

La fonction prévue à cet effet commence par rechercher les parenthèses du plus haut niveau (dans notre exemple, le pointeur se placera sur le 1 du sinus). La fonction effectuera ensuite tous les calculs compris entre la parenthèse ouvrante trouvée et la première parenthèse fermante rencontrée.

Ces calculs seront bien entendu effectués en respectant les priorités des signes et fonctions. A chaque fois qu'une opération est effectuée, l'opération est remplacée par le résultat de cette dernière. La fonction reclassée assure cette fonction. Une fois tous les calculs effectués au sein des parenthèses, celles-ci deviennent inutiles et sont supprimées.

Toutes ces opérations se répètent jusqu'à ce qu'il n'y est plus de parenthèses. La boucle est effectuée alors une dernière fois pour obtenir le résultat final.

Dans notre exemple, cela donnerait :

```
4+12*(0-2+sin(1)^2/4)E
4+12*(0-2+sin1^2/4)E
4+12*(0-2+0.841^2/4)E
4+12*(0-2+0.656/4)E
4+12*(0-2+0.164)E
4+12*(-1.836)E
4+12*-1.836E
4-22.032E
18.032E
```

Suppression des parenthèses autour du 1 :

— calcul du sinus,
— calcul de la puissance,
— calcul de la division,
— calcul du + et du −.

Suppression des parenthèses inutiles :

— calcul de la multiplication,
— calcul du −,
— résultat.

On s'aperçoit que expr[0] contient le résultat.

Une opération se passe comme suit pour l'expression suivante : 12 + 15.3.

On a :

sig	expr
	12
+	0
	15.3
E	

Lorsque sig[1] = '+' dans notre exemple, la machine effectue l'opération suivante : expr[1] = expr[0] + expr[2].

Soit :

sig	expr
	12
+	27.3
	15.3
E	0

Le reclassement s'effectue:

sig	expr
	27.3
E	0

Le résultat remplace la première variable du calcul, le signe est supprimé, et la suite du calcul écrase la deuxième variable du calcul.

Mettons maintenant le calcul suivant : 4+2*exp1.

On a :

```
sig  expr      sig  expr      sig  expr      sig  expr
      4              4              4              9.428
+     0        +     0        +     0        E     0
      2  ==>         2     ==>      5.428  ==>
*     0        *     0        E     0
e     0              2.714
x     0        E     0
p     0
      1
E     0
```

Le programme suit exactement ce raisonnement, mais peut le suivre à une échelle beaucoup plus complexe que là.

Dans le cas des équations, l'inconnu x apparaît.

Lorsqu'elle est détectée par le programme, ce dernier demande le domaine de définition sur lequel les calculs vont porter. Ce domaine se définit en 3 éléments : le minimum, le maximum et la précision (pas).

Ensuite la fonction va remplacer les inconnus x se trouvant dans l'équation par minimum et effectuer le calcul. Dans la boucle, à minimum est ajoutée précision jusqu'à maximum. Les résultats sont affichés au fur et à mesure.

Exemple : 4x+2, minimum=1, maximum=4, précision=1.

```
sig  expr      sig  expr      sig  expr      sig  expr
      4              4              4              6
*     0  ==>   *     0  ==>   +     0  ==>   E     0
x     0        x     1              2
+     0        +     0        E     0
      2              2
E     0        E     0
```

L'opération sera de nouveau effectuée avec x=2,3,4. La fonction remplace les x du tableau des signes sauf ceux contenus dans les fonctions exp et exc bien entendu.

Le programme ne travaille que sur des longs floats, ce qui lui confère une grande précision. Il est en outre aisé de stocker les résultats des équations dans un tableau si l'on souhaite tracer la droite ou la courbe.

Toutes les erreurs implicites potentielles sont traitées dans la phase des calculs, à savoir logarithme négatif ou nul, racines négatives et division par 0.

Remarques

Ce programme est intéressant dans la mesure où il permet d'appréhender les problèmes liés à la création d'un langage, à un niveau très basique il est vrai. Mais on retrouve la nécessité d'interpréter la saisie d'un utilisateur pour la rendre compréhensible (non pas pour la machine mais pour le programme, ce dernier se chargeant de le faire comprendre à la machine).

Il y a des variables, des signes, une ponctuation, des fonctions, des erreurs, des gestions de tableaux comprenant reclassement, substitution, insertion, etc... (comme la stack). Tout ce qui fait le piment d'un langage. Néanmoins, je n'aurais pas la prétention de comparer ce programme à un langage (même au basic). Mais c'est un exercice qui n'est pas dénué d'intérêt.

```
                    CALCULATRICE v4.00 (18/06/91)

Bibliothèque des fonctions :

Syntaxe                                  Descriptif

+,-,*,/ : Les opérandes classiques, les mises en facteur sont automatiques
^ : Exposant , 2^3 <Enter> résultat = 8; 27^(1/3) : racine cubique de 27 = 3
exp : exponentiel, exp1 = exp(1) = (exp1) = 2.74[....]
log : Logarithme népérien, ln10 = ln(10) etc...
sqr : racine carrée, sqr4 = sqr(4) = sqr(8/2)
sin : sinus, sin1 = sin(1)...
cos : cosinus
tan : tangente
sinh : sinus hyperbolique
cosh : cosinus hyperbolique
tanh : tangente hyperbolique
asin : arc sinus
acos : arc cosinus
atan : arc tangente
abs : Valeur absolue, abs(2-3) = 1
exc : arrondit par excès, exc(2.5) = 3
def : arrondit par défaut, def(2.5) = 2

Appuyer sur une touche

Saisissez l'expression : ((sin1)^2+(cos1)^2)exp(5)+log2

EXPRESSION TELLE QUE COMPRISE PAR L'ORDINATEUR :
((sin1)^2+(cos1)^2)*exp(5)+log2

Voulez-vous voir les calculs intermédiaires ? (o/n)

((sin1)^2+(0.5403)^2)*exp5+log2
((0.84147)^2+0.5403^2)*exp5+log2
(0.70807+0.5403^2)*exp5+log2
(0.70807+0.29193)*exp5+log2
(1)*exp5+log2
1*148.41+log2
1*148.41+0.69315
148.41+0.69315
149.11

 Le resultat est : 149.106306283137

 Voulez-vous effectuer une autre opération ? (o/n) :


Saisissez l'expression : (log(x^2)+x)exp(x)+2

EXPRESSION TELLE QUE COMPRISE PAR L'ORDINATEUR :
(log(x^2)+x)*exp(x)+2

 Equation y=(log(x^2)+x)*exp(x)+2

 Domaine de définition :

 Minimum : -2

 Maximum : 3

 Précision : 1


(log(-2^2)+-2)*exp(-2)+2 = 1.91694397353825
(log(-1^2)+-1)*exp(-1)+2 = 1.63212055882856
(log(0^2)+0)*exp(0)+2 = Indéterminé ou Impossible
(log(1^2)+1)*exp(1)+2 = 4.71828182845904
(log(2^2)+2)*exp(2)+2 = 27.0215190018074
(log(3^2)+3)*exp(3)+2 = 106.389046146185


Appuyez sur une touche ...
```

```
/*************************************************************************/
/*                                                                       */
/*            PROGRAMME CALCULATRICE v4.00 (18/06/91)                    */
/*                                                                       */
/*                      Auteur : Lefouet laurent                         */
/*                                                                       */
/*                           Langage C 2.00                              */
/*                                                                       */
/*************************************************************************/

#include <stdio.h>
#include <conio.h>
#include <math.h>

/* Variables globales */
/* sig[1000] : tableau contenant les signes de l'expression.
   expr[1000] : tableau contenant les valeurs.
   cpo et cpf : nb de parenthèses ouvertes, nb parenthèses fermées.
   dest : Position dans sig[1000] de la parenthèse ouverte de plus
          haut niveau.
   err6,err7,err4 : test des erreurs sur racines,log,division par zero.
   */

char rep=0;
int sig[1000],b=0,cpo=0,cpf=0,dest=0,pp=0,err6=0,err7=0,err4=0,inc=0;
long float expr[1000];


void presentation ()
{
 clrscr();
 cprintf ("\n\nBibliothèque des fonctions :");
 cprintf ("\n\nSyntaxe                                    Descriptif");
 cprintf ("\n\n+,-,*,/ : Les opérandes classiques, les mises en facteur sont au
tomatiques");
 cprintf ("\n^ : Exposant , 2^3 <Enter> résultat = 8; 27^(1/3) : racine cubique
de 27 = 3");
 cprintf ("\nexp : exponentiel, exp1 = exp(1) = (exp1) = 2.74[....]");
 cprintf ("\nlog : Logarithme népérien, ln10 = ln(10) etc...");
 cprintf ("\nsqr : racine carrée, sqr4 = sqr(4) = sqr(8/2)");
 cprintf ("\nsin : sinus, sin1 = sin(1)...");
 cprintf ("\ncos : cosinus");
 cprintf ("\ntan : tangente");
 cprintf ("\nsinh : sinus hyperbolique");
 cprintf ("\ncosh : cosinus hyperbolique");
 cprintf ("\ntanh : tangente hyperbolique");
 cprintf ("\nasin : arc sinus");
 cprintf ("\nacos : arc cosinus");
 cprintf ("\natan : arc tangente");
 cprintf ("\nabs : Valeur absolue, abs(2-3) = 1");
 cprintf ("\nexc : arrondit par excès, exc(2.5) = 3");
 cprintf ("\ndef : arrondit par défaut, def(2.5) = 2");
 cprintf ("\n\nAppuyer sur une touche");
 gotoxy (25,1);
 cprintf ("CALCULATRICE v4.00 (18/06/91)");
 rep=getch();
}

/* Affiche l'expression telle que l'ordinateur l'a comprise.
     ""       ""          au fur et à mesure des calculs. */

void afcalculs ()

{
 int y=0;
     do {
     if (sig[y]==0 || (strchr("cstaled",sig[y])!=NULL && expr[y]!=0))
      {cprintf ("%.4g",expr[y]);
       sig[y]=0;
       }
     else cprintf ("%c",sig[y]);
     y++;
     } while (sig[y]!='E');
     y=0;
}

/* Fonction qui remet à jour l'expression à chaque fois qu'un calcul est
   réalisé. Ce reclassement supprime de l'expression la ou les valeurs qui
   ont servis à un calcul ainsi que les signes ou fonctions qui ont été
   réalisée. En outre, elle resserre l'expression autour du résultat
   calculé. Le paramètre nb permet de reclasser correctement l'expression
   en fonction des éléments a supprimer (sin,cos... = 3, asin,acos... = 4,
     ^,+,-,*,/ = 2); */

int reclasse (int nb)
{
inc=0;
                        expr[b-nb]=expr[b];
                        inc=b+1;
                        do { inc++;
                             expr[inc-(nb+1)]=expr[inc];
                             sig[inc-(nb+1)]=sig[inc];
                        } while (sig[inc]!='E');
                          b=b-(nb+1);
                          /* affiche l'expression reclassée si les résultats
                           intermédiaires sont demandés */
                          if (rep=='o' || rep=='O') { afcalculs();
                                                     cprintf ("\n");
                                                     }
```

```
                        return (b);
 }

/* cette fonction effectue les calculs en tenant compte des priorités
   et des parenthèses */

void calcul ()
{
do {
b=0,dest=0;
/* recherche du plus haut niveau de parenthèse */
do {
    if (sig[b]==40) dest++;
    b++;
} while (dest<cpo);
pp=b;
do {
    /* calcul de la puissance x^y */
    if (sig[b]=='^')  { expr[b]=pow (expr[b-1], expr[b+1]);
                        b=reclasse (1);
                        }
    /* calcul de la valeur absolue abs x */
    if (sig[b]=='s' && sig[b-1]=='b' && sig[b-2]=='a')
                      { expr[b]=fabs (expr[b+1]);
                        b=reclasse (2);
                        }


 /* test si x n'est pas négatif ou nul après une fonction log */
 if (sig[b]=='g' && sig[b-1]=='o' && sig[b-2]=='l' && expr[b+1]<=0)
                                    { err7++;
                                      break;}
 /* Calcul la fonction log x */
 if (sig[b]=='g' && sig[b-1]=='o' && sig[b-2]=='l')
                  { expr[b]=log (expr[b+1]);
                    b=reclasse (2);
                    }

 /* test si x n'est pas négatif après une fonction sqr */
 if (sig[b]=='r' && sig[b-1]=='q' && sig[b-2]=='s' && expr[b+1]<0)
                                    { err6++;
                                      break;}
 /* Calcul la racine carrée de x : sqr x */
 if (sig[b]=='r' && sig[b-1]=='q' && sig[b-2]=='s')
                  { expr[b]=sqrt (expr[b+1]);
                    b=reclasse (2);
                    }

 /* Arrondit par excès */
 if (sig[b]=='c' && sig[b-1]=='x' && sig[b-2]=='e')
                  { expr[b]=ceil (expr[b+1]);
                    b=reclasse (2);
                    }

 /* Arrondit par défaut */
 if (sig[b]=='f' && sig[b-1]=='e' && sig[b-2]=='d')
                  { expr[b]=floor (expr[b+1]);
                    b=reclasse (2);
                    }

 /* Calcul l'exponentiel exp de x */
 if (sig[b]=='p' && sig[b-1]=='x' && sig[b-2]=='e')
                  { expr[b]=exp (expr[b+1]);
                    b=reclasse (2);
                    }

/*  Calcul du sinus hyperbolique */
if (sig[b]=='h' && sig[b-1]=='n' && sig[b-2]=='i' && sig[b-3]=='s')
                 {
                   expr[b]=sinh (expr[b+1]);
                   b=reclasse (3);
                   }

/* Calcul du cosinus hyperbolique */
if (sig[b]=='h' && sig[b-1]=='s' && sig[b-2]=='o' && sig[b-3]=='c')
                 {
                   expr[b]=cosh (expr[b+1]);
                   b=reclasse (3);
                   }

/* Calcul de la tangente hyperbolique */
if (sig[b]=='h' && sig[b-1]=='n' && sig[b-2]=='a' && sig[b-3]=='t')
                 {
                   expr[b]=tanh (expr[b+1]);
                   b=reclasse (3);
                   }

/* calcul du sinus et de l'arc sinus */
if (sig[b]=='n' && sig[b-1]=='i' && sig[b-2]=='s' && sig[b+1]!='h')
if (sig[b-3]=='a')

                     { expr[b]=asin (expr[b+1]);
                       b=reclasse (3);
                      }
         else
                     { expr[b]=sin (expr[b+1]);
                       b=reclasse (2);
                       }
```

```
      /* calcul du cosinus et de l'arc cosinus */
     if (sig[b]=='s' && sig[b-1]=='o' && sig[b-2]=='c' && sig[b+1]!='h')
         if (sig[b-3]=='a')
                        { expr[b]=acos (expr[b+1]);
                          b=reclasse (3);
                        }
           else
                        { expr[b]=cos (expr[b+1]);
                          b=reclasse (2);
                          }

      /* calcul de la tangente et de l'arc tangente */
     if (sig[b]=='n' && sig[b-1]=='a' && sig[b-2]=='t' && sig[b+1]!='h')
            if (sig[b-3]=='a')
                        { expr[b]=atan (expr[b+1]);
                          b=reclasse (3);
                          }
           else
                        { expr[b]=tan (expr[b+1]);
                          b=reclasse (2);
                          }
         b++;
/* Tous ces calculs sont effectuées jusqu'à ce que l'on trouve la
   parenthèse fermante ou la fin de l'expression notée 'E' */
} while (sig[b]!=41 && sig[b]!=69);
b=pp;
do {
    /* Calcul de la multiplication */
    if (sig[b]==42)  { expr[b]=expr[b-1]*expr[b+1];
                       b=reclasse (1);
                       }

    /* Test si la valeur suivant une division n'est pas nulle */
    if (sig[b]==47 && expr[b+1]==0) { err4++;
                                      break;}

    /* Calcul de la division */
    if (sig[b]==47)  { expr[b]=expr[b-1]/expr[b+1];
                       b=reclasse (1);
                       }
    b++;
} while (sig[b]!=41 && sig[b]!=69);
b=pp;
do {
if (err4>0) break;

    /* Calcul de l'addition */
    if (sig[b]==43)  { expr[b]=expr[b-1]+expr[b+1];
                       b=reclasse (1);
                       }

    /* Calcul de la soustraction */
    if (sig[b]==45)  { expr[b]=expr[b-1]-expr[b+1];
                       b=reclasse (1);


      b++;
} while (sig[b]!=41 && sig[b]!=69);
/* suppression des parenthèses inutiles */

if (cpo>0) {
b=pp-1;
expr[b]=expr[b+1];
sig[b]=0;
inc=b+2;
do { inc++;
expr[inc-2]=expr[inc];
sig[inc-2]=sig[inc];
} while (sig[inc-2]!='E');}
if (cpo>=0)
{
cpo--;
cpf--;
}
} while (cpo>=0);
}

/* Cette fonction permet d'obtenir le tableau des valeurs
   de l'équation saisie, uniquement dans le cas où l'expression
   contient des inconnus x */

void fonction ()
{
 int i=0,j=0,sig2[1000];
 long float expr2[1000];
 long float mn=0,mx=0,pre=0;

 for (i=0;i<=1000;i++)
 sig2[i]=expr2[i]=0;
 i=0;
 do
{
 sig2[i]=sig[i];
 expr2[i]=expr[i];
 i++;
 } while (sig[i-1]!='E');
cprintf ("\n\n Equation y=");
afcalculs();
```

```
cprintf ("\n\n Domaine de définition :");
cprintf ("\n\n Minimum : ");
scanf ("%lf",&mn);
cprintf ("\n Maximum : ");
scanf ("%lf",&mx);
cprintf ("\n Précision : ");
scanf ("%lf",&pre);
cprintf ("\n\n");
j=0;
do {
i=cpo=cpf=0;
if (mn>(-2e-12) && mn<(2e-12)) mn=0;
/* remplace x dans l'equation par une valeur comprise entre
mn et mx avec un pas de 'pre' */
do {
 sig[i]=sig2[i];
 expr[i]=expr2[i];
 if (sig[i]=='x' && sig[i-1]!='e' && (sig[i+1]!='p' || sig[i+1]!='c')) {

                      sig[i]=0;
                      expr[i]=mn;
                      }
  if (sig[i]==40) cpo++;
  if (sig[i]==41) cpf++;
    i++;
  } while (sig2[i-1]!='E');
  i=0;

  /* affichage du résultat de l'equation */

  afcalculs();
  calcul();
  if (err4!=0 || err6!=0 || err7!=0)
  {
   cprintf (" = Indéterminé ou Impossible\n");
   err4=err6=err7=0;
   }
   else
   cprintf (" = %.14g\n",expr[0]);
   if (j>=20) {
               cprintf ("\nAppuyez sur une touche ...\n\n");
               rep=getch();
               j=0;
             }
  mn=mn+pre;
  j++;
  } while (mn<=mx);
  if (j>1)
  { cprintf ("\n\nAppuyez sur une touche ...");
    rep=getch();}
}

main ()
{ char t[1000],ky;
  int a=0,x=0,y=0,f=0,err1=0,err2=0,stop=0;
  int demar=0,err3=0,err5=0,mem=0,n=0,z=0;
do {
do {
  stop=0,f=0,a=0,cpo=0,cpf=0,y=0,x=0;
  do { expr[f]=sig[f]=t[f]=0;
       f++;
  } while (f<=1000);
  f=0;
  if (demar==0) presentation();
  demar=1;
  clrscr ();
  cprintf ("\nSaisissez l'expression : ");
  /* Boucle de saisie avec émulation du <DEL> qui permet mieux qu'un
     scanf de ne pas être limitée par 255 caractères */
  do {
      t[a]=getche();
      if (t[a]=='x')
         if (t[a-1]>47 && t[a-1]<58) {
                                      t[a+1]='x';
                                      t[a]='*';
                                      a++;
                                      }
      if (t[a]==8)
        { if (a>0)
                    {t[a]=0;
                     cprintf (" \b");

                    a--;
                    if (t[a]=='x') {
                                    t[a]=0;
                                    a--;
                                    }
                    t[a]=0;
                    }
          else {
                t[a]=0;
                gotoxy (26,2);
                printf ("\a");
                }
                }
     if (t[a]==32) a--;
     if (t[a]!=0) a++;
 } while (t[a-1]!=13);
 t[a-1]=0;
 a=0;
```

```
        /* Boucle de décomposition de la chaine de caractères t[1000] contenant
           l'expression en une expression 'compréhensible' par la fonction calcul
           Cette boucle sépare les signes et fonctions des valeurs, convertit les
           codes ascii des nombres en valeur recomposée.
           Elle introduit le signe '*' lors des mises en facteurs etc.. */

        do {
             /* Dans le cas où l'expression contient un élément du type (-2 ....
                , l'algorithme introduit un 0 entre la parenthèse et le signe car
                parenthèse - 2 ne signifie rien pour la fonction calcul => erreur,
                alors que 0-2 egal bien -2 . */
             if ((t[a-1]=='(' || a==0) && (t[a]=='+' || t[a]=='-')) { sig[y+1]=sig[y];
                                                                      sig[y]=expr[y]=expr[y+1]=0;
                                                                      y++;}

             /* Test des mises en facteurs.
                introduit le signe '*' dans le cas où l'expression contient des
                 ')(' , ')asin' , ')5' */

             if (t[a-1]==')' && (t[a]=='(' || t[a]=='a' || t[a]=='c' || t[a]=='s'
                             || t[a]=='t' || t[a]=='l' || t[a]=='e' || t[a]=='d'
                             || t[a]=='x' || (t[a]>47 && t[a]<58)))
                                          { sig[y+1]=sig[y];
                                          sig[y]='*';
                                          expr[y]=expr[y+1]=0;
                                          y++;}

             /* Algorithme de reconstitution de la partie entière d'un nombre.
                La variable mem sert à savoir si l'on se trouve juste après
                un point décimal ou pas. On reconstitue la partie entière d'un
                nombre à partir des valeur ascii des chiffres saisis, cette
                opération est effectuée pour les chiffres encadrés entre deux
                signes. x permet de savoir combien il y a de chiffre entre
                le dernier signe rencontré et le signe suivant. Ex :
                +123). On a contée 3 chiffres à partir du '+' jusqu'à la
                parenthèse. on a donc t[a-3]=49 (code ascii de 1) t[a-2]=50
                et t[a-1]=51. Pour avoir 123, il faut faire (49-48)*10^2 +
                (50-48)*10^1 + (51-48)*10^0 = 1*100 + 2*10 + 3*1 = 100+20+3 =
                123, CQFD. On soustrait 48, code ascii du 0 pour retrouver la
                valeur du chiffre . */

            if ( mem==0 && (t[a]=='(' || t[a]==')' || t[a]=='+' || t[a]=='-'
                || t[a]=='/' || t[a]=='*' || t[a]=='.' || t[a]=='^' || t[a]==0))
            {   if ((t[a-1]>=48 && t[a-1]<=57) || t[a]!=13)

                { if (t[a]==94) t[a]=1;
                  do {
                  expr[y]=expr[y]+((t[a-(x)]-48)*pow (10, (x-1)));
                  x--;
                  } while (x>0);
                  if (t[a]==1) t[a]=94;
                x=-1;
                n=0;
                if (expr[y]>=0 && t[a]!='.') y++;
                if (t[a]!='.') { sig[y]=t[a];
                                 expr[y]=0;
                                 y++;}
                                 }}

           /* Calcul de la partie décimal d'un nombre. Ce calcul est effectué
           sur le même principe, sauf que l'on compte les chiffres à partir
           du dernier point décimal rencontré, et la reconstitution s'effectue
           à l'aide des puissances négatives de 10. */

           if ( mem==1 && (t[a]=='(' || t[a]==')' || t[a]=='+' || t[a]=='-'
               || t[a]=='/' || t[a]=='*' || t[a]=='^' || t[a]==0))
           { if ((t[a]!=13) || (t[a-1]>=48 && t[a-1]<=57))
                { n=-1;
                  z=x;
                  do { expr[y]=expr[y]+((t[a-(x)]-48)*pow (10, (n)));
                  n--;
                  x--;
                } while (n>=(-z));
                x=-1;
                if (expr[y]>=0) y++;
                sig[y]=t[a];
                expr[y]=0;
                y++;
                mem=0;}}

           /* Test si un nombre ne contient pas deux points décimaux */
           if (t[a]=='.' && mem==1) err5++;
           /* Si l'on rencontre un point, on sait qu'il va falloir calculer
              une partie décimale => mem=1 */
           if (t[a]=='.') mem=1;
           /* on stock les noms des fonctions dans le tableau sig[] */
           if (strchr ("scionatbhlgepxdfqr",t[a])!=NULL && t[a]!=0)
                   { x=-1;
                     sig[y]=t[a];
                     y++;}
           /* Introduit le signe '*' dans le cas des factorisations
             du type '5(' */
           if ((t[a-1]>=48 && t[a-1]<=57) && t[a]=='(') { sig[y]=t[a];
                                                         sig[y-1]='*';
                                                         y++;}
           a++;
           x++;
       } while (t[a-1]!=0);
     /* Ajoute un 'E' pour marquer la fin de l'expression */
   if (sig[y-1]!=40 && sig[y-1]!=41) sig[y-1]='E';
      else sig[y]='E';
```

```
/* Recherche les erreurs de saisies ou mathématiques */
b=0;
do
   { if (sig[b]==40) cpo++;
     if (sig[b]==41) cpf++;
     if (sig[b]==40 && (sig[b+1]=='*' || sig[b+1]=='/' || sig[b+1]=='^'))
                err1++;
     if (sig[b]==41 &&
        (sig[b-1]=='*' || sig[b-1]=='/' || sig[b-1]=='+' || sig[b-1]=='-' || sig
[b-1]=='^'))
        err2++;
     if (sig[b]!=0)
      if (sig[b]!=40 && sig[b]!=41 && sig[b]!='x')
      if (strchr ("sincoatbhlgexpdfqr",sig[b+1])==NULL)
        { if (sig[b+1]!=40 && sig[b+1]!=41 && sig[b+1]!=0)  err3++;}
     if (sig[b]=='/' && sig [b+1]==0 && expr[b+1]==0) err4++;
     b++;
     } while (sig[b-1]!='E');
     if (cpo>cpf) {
                    cprintf ("\nErreur 8 : Il manque %d parenthese(s) fermante(s
)...",(cpo-cpf));
                    stop=1;}
     if (cpo<cpf) {
                    cprintf ("\nErreur 9 : Il manque %d parenthese(s) ouvrante(s
)...",(cpf-cpo));
                    stop=1;}
     if (err1>0) {
                    cprintf ("\nErreur 1 : %d Erreur(s) du type '(^ (* (/'...",e
rr1);
                    stop=1;}
     if (err2>0) {
                    cprintf ("\nErreur 2 : %d Erreur(s) du type '^) -) +) /) *)'
...",err2);
                    stop=1;}
     if (err3>0) {
                    cprintf ("\nErreur 3 : %d Erreur(s), deux signes sont consec
utifs ...",err3);
                    stop=1;}
     if (err4>0) {
                    cprintf ("\nErreur 4 : %d Erreur(s), division par zero ...",
err4);
                    stop=1;}
     if (err5>0) {
                   cprintf ("\nErreur 5 : %d Erreur(s), Syntaxe des points decim
aux...",err5);
                    stop=1;}
     cprintf ("\n\nEXPRESSION TELLE QUE COMPRISE PAR L'ORDINATEUR :\n");
     y=0;
     afcalculs();
     if (err1!=0 || err2!=0 || err3!=0 || err4!=0 || err5!=0 || cpo<cpf
      || cpo>cpf)
     {
     cprintf ("\n\nAppuyez sur une touche...");
     ky=getch();
     }
     b=0,x=0,y=0,err1=0,err2=0,err3=0,err4=0,err5=0,err6=0,err7=0,ky=0,dest=0;
} while (stop!=0);

/* Détecte s'il s'agit d'un calcul ou d'une équation à résoudre */

do {
   if (sig[x]=='x' && sig[x-1]!='e' && (sig[x+1]!='p' || sig[x+1]!='c')) ky=1;
   x++;
   } while (sig[x-1]!='E');
   if (x<=2) ky=2;
   if (ky==1) fonction ();
   else
 {
    cprintf ("\n\nVoulez-vous voir les calculs intermédiaires ? (o/n)");
      rep=getch ();
      printf ("\n\n");
    if (ky==0) calcul();
    x=0;

 /* Certaines expressions sont implicites, et l'on ne peut pas
    toujours détecter que telle opération va donner 0 ou un nombre
    négatif. C'est la raison pour laquelle des tests sont effectués pendant
    les calculs afin d'éviter un plantage. */
 if (err4>0) {
                    cprintf ("\n\nErreur 4 : %d Erreur(s), division par zero ..
.\n",err4);
                    }
 else
 if (err6>0) {
                    cprintf ("\n\nErreur 6 : %d Erreur(s), racine négative ...\
n",err6);
                    }
 else
 if (err7>0) {
                    cprintf ("\n\nErreur 7 : %d Erreur(s), logarithme négatif o
u nul...\n",err7);
                    }
 else {
 cprintf ("\nLe resultat est : %.14g ",expr[0]);}
 }
 cprintf ("\n\nVoulez-vous effectuer une autre opération ? (o/n) :");
 ky=getche ();
 err4=err6=err7=rep=0;
 } while (ky=='o' || ky=='O');
 }
```

N° 1 AUX ETATS-UNIS
EN VERSION FRANCAISE

Réalisez vous-même vos applications personnalisées sans taper une seule ligne de programme :

Créez des applications relationnelles de qualité et distribuez-les librement. Base de données, multi-fichiers indexés, écran multifenêtres, menus, tableaux déroulants, masques de saisie, édition de toute nature, champs calculés, conditionnels, totaux, etc...

C'est facile ! C'est étonnant !

Matériel requis : micro-ordinateur compatible PC et système d'exploitation MS-DOS, 512 K RAM, 2 MO sur disque.

Spécifications techniques : Multifichiers, 252 fichiers par application, 251 clés par fichier, 255 champs par enregistrement. Longueur enregistrement jusqu'à 64 Ko.

Taille des fichiers : illimitée.

Mode automatique : crée une application totale en quelques secondes (menus, tableaux déroulants, masques de saisie, états) à partir de la simple définition du fichier.

Aide à l'utilisation : intégrée (plus de 400 écrans).

Conversion de fichiers : import/export avec fichiers dBase, DIF, ASCII délimité.

EN CADEAU

Huit applications types, déjà développées, utilisables telles quelles ou modifiables. (Inventaire, gestion de club, mailing, facturation, répertoire téléphonique, suivi commercial, gestion scolaire).

BIONIX SOFTWARE
32, rue Molitor 75016 Paris
Tél : (33) (1) 40 71 09 36
Fax : (33) (1) 46 51 24 26

Pour la Suisse :
IHS Distribution S.A.
En Budron A 16
1052 Le Mont / Lausanne
Tél : (021) 652 34 36
Fax : (021) 652 34 52

Pour la Belgique :
Nasa Informatique S.A.
Rue Godefroid, 30 5000 Namur
Tél : (081) 22 40 77
Fax : (081) 23 19 35

LE PRIX DE LA PERFORMANCE: 990,00 F HT

PRIX DE LANCEMENT

PRIX TARIF : 2390,00 F HT.

PHOTO COMSTOCK EDITION SPECIALE

CLARION Personnel

LE GENERATEUR D'APPLICATION UNIVERSEL

BON DE COMMANDE à remplir et à retouner à BIONIX.

Je désire recevoir CLARION Personnel au prix de lancement de 990, 00 Francs HT plus la TVA et ma participation aux frais d'envoi soit 1249,14 francs TTC.
Veuillez trouver ci-joint mon chèque de réglement à l'ordre de BIONIX.

❑ Disquettes 3 1/2 pouces
❑ Disquette 5 1/4 pouces

Signature

Nom _____ Société _____

Fonction _____ Tél _____

Adresse _____

_____

Code Postal |_|_|_|_|_| Ville _____

MARLO le réflexe micro

ACER 1120/SX

80386SX-20 — 2 Mo RAM — 1 disque dur 40 Mo — VGA couleurs — Garantie 1 an sur site
Prix MARLO : 13 500 F

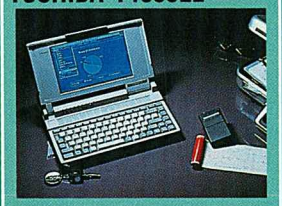
SHARP PC6661

80386SX-20 MHz — 2 Mo RAM — 1 lecteur 3''1/2 — Disque dur 60 Mo — Écran LCD VGA — DOS — Poids : 2,7 kg
Prix MARLO : 17 990 F

COMPAQ LTE 386S/20

80386SX-20 MHz — 2 Mo RAM — 1 lecteur 3''1/2 — Disque 30 Mo — Écran LCD VGA — Poids : 3,4 kg
Prix MARLO : 21 990 F
Version 60 Mo : 23 790 F

TOSHIBA T1000LE



80C86 à 9,54 MHz — 1 Mo RAM — Disque dur 20 Mo — LCD 640×400 — Poids : 3 kg — DOS
Prix MARLO : 10 500 F

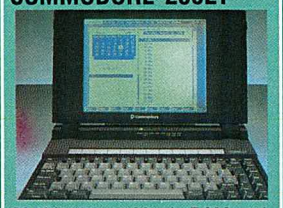
TOSHIBA T2000SXe

80386SX-20 — 2 Mo RAM — Disque dur 40 Mo — LCD VGA — Poids : 2,5 kg — DOS
Prix MARLO : 21 990 F
Version 60 Mo : 23 490 F

COMMODORE DT286-12

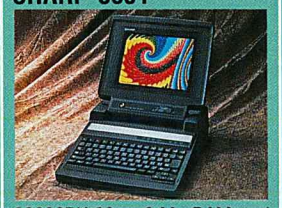
80286-12 — 1 Mo RAM — 1 lecteur 3''1/2 — Disque dur 40 Mo — Écran VGA couleurs 14'' — MSDOS — WORKS 2 + GEOWORKS
Prix MARLO : 6 300 F

COMMODORE 286LT



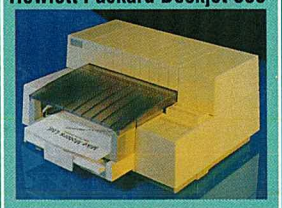
80286-12 — 1 Mo de RAM — 1 lecteur 3''1/2 — Disque dur 20 Mo — Écran LCD VGA — DOS — Housse-Works 2 — Autonome
Prix MARLO : 9 900 F

SHARP 8501



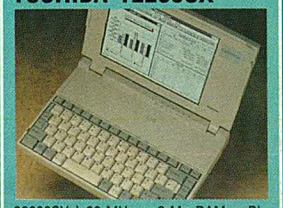
80386DX-20 — 2 Mo RAM — 1 lecteur 3''1/2 — Disque dur 100 Mo — Écran LCD VGA couleurs — DOS
Prix MARLO : 43 800 F

Hewlett Packard Deskjet 500



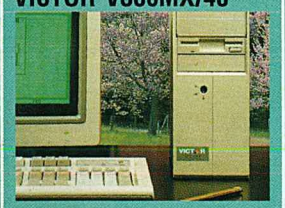
Imprimante à jet d'encre — 2 à 3 pages/minute — Garantie 3 ans dont 1 an sur site.
Prix MARLO : 3 400 F

TOSHIBA T2200SX



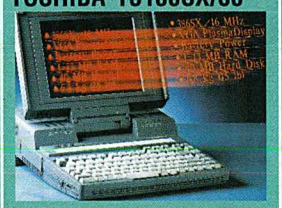
80386SX à 20 MHz — 2 Mo RAM — Disque 40 Mo — Écran LCD VGA — DOS — Poids : 2,5 kg — Autonomie : 4 heures.
Prix MARLO : 23 490 F
Version 60 Mo : 24 490 F

VICTOR V386MX/40



80386-16 — 2 Mo RAM — Disque dur 52 Mo — Écran VGA couleurs — DOS — Windows
Prix MARLO : 13 500 F

TOSHIBA T3100SX/80



80386SX-16 — 1 Mo RAM — Disque dur 80 Mo — Plasma VGA — Poids : 6,8 kg — DOS
Prix MARLO : 20 990 F

COMPAQ 386/25e

80386-25 — 4 Mo RAM — Disque dur de 60 Mo - Écran VGA couleurs
Prix MARLO : 29 700 F

TOSHIBA T3200SXC

80386SX-20 — 1 Mo RAM — Disque dur de 120 Mo — LCD VGA couleurs — Poids : 7,9 kg — DOS
Prix MARLO : 43 990 F

TOSHIBA T5200

80386-20 — 2 Mo RAM — Disque dur 100 Mo — Plasma VGA — Poids : 8,5 kg — DOS
Prix MARLO : 26 990 F
Version 200 Mo : 30 990 F

STAR LS8 III

8 ppm — 1 Mo RAM — Garantie 1 an sur site
Prix MARLO : 10 990 F
Version PostScript : 16 500 F

STAR LS4

4 ppm — 1 Mo RAM — Garantie 1 an sur site
Prix MARLO : 6 990 F
Version PostScript : 9 790 F

CANON LPB4+

4 pages minute — 512 Ko RAM — 300 Dpi — Cassette 50 feuilles — Garantie 1 an sur site
Prix MARLO : 5 990 F

Hewlett Packard Série III

8 ppm — 1 Mo de RAM — Garantie 1 an sur site
Prix MARLO : 11 990 F
Version PostScript : 14 790 F

CHICONY NB

80386SX-20 — 1 Mo RAM — Disque dur 40 Mo — LCD VGA — Poids : 3,4 kg — DOS — Garantie 3 ans
Prix MARLO : 12 500 F
Version 60 Mo : 14 500 F

TOSHIBA T2000

80286 à 12 MHz — 1 Mo RAM — Disque 20 Mo — Écran LCD VGA — DOS — Poids : 3,1 kg
Prix MARLO : 14 990 F
Version 40 Mo : 15 890 F

Hewlett Packard Série III P

4 pages minute — 1 Mo RAM — 300 Dpi — Garantie 1 an sur site.
Prix MARLO : 7 990 F
Version PostScript : 11 195 F

SHARP PC6240

80C286 à 12 MHz — 1 Mo RAM — LCD VGA — Disque 40 Mo — Poids : 2,2 kg — DOS
Prix MARLO : 12 500 F

Financement Crédit Universel

MARLO

13, boulevard Ch.-de-Gaulle - 92700 COLOMBES

Tél. : (1) 47 81 20 57

Fax : (1) 47 80 59 63

OUVERT DU LUNDI AU SAMEDI

Tous nos prix s'entendent hors taxes T.V.A. 18,6 % en sus. Prix modifiables sans préavis.